On the Complexity of Computing Treebreadth

Guillaume Ducoffe, Sylvain Legay, Nicolas Nisse

On the complexity of computing treebreadth *†

Guillaume Ducoffe‡[1,2], Sylvain Legay[3], and Nicolas Nisse[1]

[1]Université Côte d'Azur, Inria, CNRS, I3S, France
[2]National Institute for Research and Development in Informatics, Romania
[3]LRI, Univ Paris Sud, Université Paris-Saclay, 91405 Orsay, France

Abstract

During the last decade, metric properties of the *bags* of tree decompositions of graphs have been studied. Roughly, the *length* and the *breadth* of a tree decomposition are the maximum diameter and radius of its bags respectively. The *treelength* and the *treebreadth* of a graph are the minimum length and breadth of its tree decompositions respectively. *Pathlength* and *pathbreadth* are defined similarly for path decompositions. In this paper, we answer open questions of [Dragan and Köhler, Algorithmica 2014] and [Dragan, Köhler and Leitert, SWAT 2014] about the computational complexity of treebreadth, pathbreadth and pathlength. Namely, we prove that computing these graph invariants is NP-hard. We further investigate graphs with treebreadth one, i.e., graphs that admit a tree decomposition where each bag has a dominating vertex. We show that it is NP-complete to decide whether a graph belongs to this class. We then prove some structural properties of such graphs which allows us to design polynomial-time algorithms to decide whether a bipartite graph, resp., a planar graph (or more generally, a triangle-free graph, resp., a $K_{3,3}$-minor-free graph), has treebreadth one.

1 Introduction

Tree decompositions [37] aim at decomposing graphs into pieces, called *bags*, organized in a tree-like manner (formal definitions are postponed to Section 1.3). Roughly, the *width* of a tree decomposition is the maximum size of its bags. A lot of work has been dedicated to compute tree decompositions with small width since such decompositions can be efficiently exploited for algorithmic purposes [8]. Computing the corresponding graph invariant, the *treewidth* of a graph G (i.e., the minimum width among all tree decompositions of G), is NP-hard [3] and no constant-approximation algorithm is likely to exist [39]. Moreover, real-life networks generally have a large treewidth [17]. These drawbacks motivated the study of other optimization criteria for tree decompositions.

In particular, the metric properties of the bags have been studied. Roughly, the *length* and the *breadth* of a tree decomposition are the maximum diameter and radius of its bags respectively.

*This work is partially supported by ANR project Stint under reference ANR-13-BS02-0007 and ANR program "Investments for the Future" under reference ANR-11-LABX-0031-01. This work was also supported by project PN 19 37 04 01 "New solutions for complex problems in current ICT research fields based on modelling and optimization", funded by the Romanian Core Program of the Ministry of Research and Innovation (MCI) 2019-2022. This work was also supported by a grant of Romanian Ministry of Research and Innovation CCCDI-UEFISCDI. project no. 17PCCDI/2018.

†Results of this paper were partially presented at the IWOCA'16 conference [25].

‡Corresponding author: guillaume.ducoffe@ici.ro

The corresponding graph parameters are the *treelength* [19] and the *treebreadth* [21] respectively. Intuitively, it follows that the treelength of a graph is between its treebreadth and twice its treebreadth. Note however that each bound may be reached (e.g., consider any complete graph, for which both treelength and treebreadth equal one; and a 4-vertex cycle for which the treebreadth and the treelength equal 1 and 2, respectively). Therefore, both parameters may behave differently.

One motivation for studying both treelength and treebreadth is that, according to recent studies, some classes of real-life networks – including biological networks and social networks – have bounded treebreadth [1]. This metric tree-likeness can be exploited in algorithms. For instance, parameters such as the metric dimension of graphs are FPT on bounded treelength (and so, bounded treebreadth) graphs [4]. Moreover, bounded treebreadth (and so bounded treelength) graphs admit a PTAS for the Traveling Salesman problem [30]. They also admit "good" additive tree-spanners [21] and compact distance labeling schemes [18]. Furthermore, the diameter and the radius of bounded treebreadth graphs can be approximated up to an additive constant in linear time [15]. In contrast to the above result, we emphasize that under classical complexity assumptions the diameter of general graphs *cannot* be approximated up to an additive constant in subquadratic time, that is prohibitive for large graphs [14]. Lastly, deciding whether a graph of treebreadth one is 3-colorable can be done in polynomial time [31]. Note that in contrast, the complexity of this problem is still open for graphs of diameter two [34], and so also for graphs of treelength two.

On the computational side, it is known that computing the treelength is NP-hard [33]. However, contrary to the treewidth, there exists a 3-approximation algorithm for computing the treelength [19]. In [21], a 3-approximation algorithm for computing the treebreadth is presented but the computational complexity of this problem is left open. Note that, because treelength and treebreadth differ by at most a factor 2 [21], any polynomial-time algorithm for computing the treebreadth, or an α-approximation algorithm for some $\alpha < 3/2$, would improve the 3-approximation algorithm for treelength [19]. Finally, the question of the computational complexity of all these parameters (treewidth, treelength and treebreadth) is open in planar graphs. This latter observation motivates the last section of this work where we give first results concerning the complexity of treebreadth in planar graphs.

A *path decomposition* of a graph is a tree decomposition where the bags are organized according to a path structure. Treelength and treebreadth have their "path counterpart", namely the *pathlength* and the *pathbreadth*. In [22], they have been shown to be useful in the design of approximation algorithms for bandwidth and line-distortion. A 2-approximation (resp., a 3-approximation) algorithm is given for computing the pathlength (resp., the pathbreadth) but the computational complexity of both problems is left open.

The main contributions of this paper are to answer the open problems of [21] and [22]. Namely, we prove that computing the treebreadth, pathlength and pathbreadth of graphs are all NP-hard problems.

1.1 Related work.

In contrast with treewidth [7], deciding whether a graph has treelength at most k is NP-complete for every fixed $k \geq 2$ [33]. However, the reduction used for treelength goes through weighted graphs and then goes back to unweighted graphs using rather elegant gadgets. It does not seem to us these gadgets can be easily generalized in order to apply to the treebreadth.

Relationship between treewidth and treelength (and so, treebreadth) has been investigated in [16]. The two parameters are uncomparable in general graphs. For instance, cycles have treewidth at most two but treelength $\lceil n/3 \rceil$, while cliques have treewidth $n-1$ but treelength equal to one [19]. However, treewidth and treelength differ by at most a constant ratio in the graphs with bounded genus and bounded isometric cycles [16]. Hence we are also motivated in this work to better understand the structure of tree decompositions with small width for bounded genus graphs, and to improve their computation.

Recently, the MINIMUM ECCENTRICITY SHORTEST-PATH problem – close to the problem of computing the pathlength and pathbreadth – has been proved NP-hard [23]. Let us point out that for every fixed k, it can be decided in polynomial time whether a graph admits a shortest-path with eccentricity at most k [23]. Our results will show the situation is different for pathlength and pathbreadth.

Last, following a preliminary version of this work [25], a new parameter called *strong treebreadth* has been introduced in [32]. Roughly, in order for a tree decomposition to have strong breadth equal to ρ, each bag must be the *complete* ρ-neighbourhood of a vertex. Deciding whether a given graph has strong treebreadth at most k is NP-complete for every $k \geq 1$ [32]. More recently, it has been proved in [24] that computing the *strong pathbreadth* is also NP-hard.

1.2 Our contributions.

On the negative side, we prove in Section 2 that computing the treebreadth is NP-hard. More precisely, we first prove that recognizing graphs with treebreadth one is NP-complete. The latter may be a bit surprising since in comparison, graphs with treelength one are exactly the chordal graphs [33], and so, they can be recognized in linear time. Our reduction has distant similarities with the one for treelength. However, it does not need any detour through weighted graphs. Then, we show that the problem of deciding whether a graph has treebreadth one is polynomially equivalent to the problem of deciding whether a graph has treebreadth at most k, for every fixed $k \geq 1$.

Next, we show that deciding if a graph has pathlength at most 2 is NP-complete even in the class of graphs with diameter (and so, pathlength) at most 3. We also show that deciding if a graph has pathbreadth at most 1 is NP-complete even in the class of graphs with radius (and so, pathbreadth) at most 2. Hence, for any $\epsilon > 0$, the pathlength and the pathbreadth cannot be approximated within a factor $\frac{3}{2} - \epsilon$ and $2 - \epsilon$ respectively unless $P = NP$.

On the positive side, we present in Section 3 polynomial-time algorithms for deciding whether a graph has treebreadth at most one, in the class of triangle-free graphs and in the class of $K_{3,3}$-minor-free graphs (note that the latter are superclasses of bipartite graphs and of planar graphs, respectively). Our main insight for triangle-free graphs is that, assuming they have no clique-separators, having treebreadth one is equivalent to having *strong* treebreadth one. Then, we can apply an algorithm from [32] which given a graph of strong treebreadth ρ, computes a tree decomposition of breadth at most ρ. For the special case of bipartite graphs, we obtain a more elegant characterization. Precisely, we prove that a bipartite graph has treebreadth one if and only if it can be clique-decomposed in *tree-convex* bipartite graphs [38]. Furthermore, while the $K_{3,3}$-minor-free graphs of treebreadth one are quite specific (in particular, we prove that they have treewidth at most 4), our algorithm for these graphs in order to decide whether they have treebreadth one is intricate and relies on structural properties of graphs with treebreadth one.

1.3 Definitions and notations

Graphs in this study are finite, simple, connected and unweighted. Given a graph $G = (V, E)$, the set $N_G(v)$ denotes the set of neighbors of $v \in V$ in G. Furthermore, let $N_G[v] = N_G(v) \cup \{v\}$. The distance $dist_G(u, v)$ between two vertices $u, v \in V$ in G is the minimum length (number of edges) of a path between u and v in G. We will omit the subscript when no ambiguity occurs.

A graph H is a *contraction-minor* of a graph G if H is obtained from G by contracting some edges. More generally, H is a *minor* of G if H is a subgraph of any contraction-minor of G. A graph G is *H-minor-free* if it does not admit H as a minor.

A *tree decomposition* $(T, \mathcal{X})$ of G is a pair consisting of a tree T and of a family $\mathcal{X} = (X_t)_{t \in V(T)}$ of subsets of V indexed by the nodes of T and satisfying:

- $\bigcup_{t \in V(T)} X_t = V$;
- for any edge $e = \{u, v\} \in E$, there exists $t \in V(T)$ such that $u, v \in X_t$;
- for any $v \in V$, $\{t \in V(T) \mid v \in X_t\}$ induces a subtree, denoted by T_v, of T.

The sets X_t are called *the bags* of the decomposition. For any $t \in V(T)$, the *diameter* of the bag X_t equals $\max_{v,w \in X_t} dist_G(v, w)$. We emphasize that the distance is the one in G (not in $G[X_t]$). The *radius* of X_t equals $\min_{v \in V} \max_{w \in X_t} dist_G(v, w)$. Equivalently, the radius of X_t is the minimum ρ such that $X_t \subseteq B_G(v, \rho) = \{u \in V \mid dist_G(u, v) \leq \rho\}$ for some $v \in V$. We point out that the vertex v in previous definition does not necessarily belong to X_t[1]. The *length* of $(T, \mathcal{X})$ is the maximum diameter of its bags, while the *breadth* of $(T, \mathcal{X})$ is the maximum radius of its bags.

The *treelength* and the *treebreadth* of G, respectively denoted by $tl(G)$ and $tb(G)$, are the minimum length and breadth of its tree decompositions, respectively. Pathlength and pathbreadth are defined similarly in the case of path decompositions, that is, when T is a path. They are respectively denoted by $pl(G)$ and $pb(G)$ in what follows. Furthermore it has been observed in [21, 22] that the four above parameters are contraction-closed invariants.

A *clique-tree* is a tree decomposition with length at most one (*i.e.*, where all the bags are cliques). A graph is *chordal* if and only if it has a clique-tree, and so, chordal graphs are exactly the graphs with treelength at most one [27, 33].

Finally, a tree decomposition is called *reduced* if no bag is included in another one. Starting from any tree decomposition, a reduced tree decomposition can be obtained in polynomial time by contracting any two adjacent bags with one contained in the other until it is no more possible to do that. Note that such a process does not modify the width, the length nor the breadth of the decomposition.

In the following we will make use of the well-known *Helly property* in our proofs: any family of pairwise intersecting subtrees in a tree has a nonempty intersection [29].

2 Hardness of treebreadth, pathlength and pathbreadth

The main result of this section is the NP-completeness of deciding whether $tb(G) \leq k$, for any fixed $k \geq 1$. We first prove that the problem is NP-complete for $k = 1$. Then, we show that the problem of deciding the treebreadth of a graph is polynomially equivalent to the problem of recognizing graphs with treebreadth one. Using similar techniques, we prove that computing pathlength, resp., pathbreadth, is NP-hard.

[1] Our notions of diameter and radius are sometimes called *weak* diameter and *weak* radius in the literature [20].

2.1 Treebreadth

We start by a structural result on graphs with treebreadth one which will be a key lemma used throughout the paper. A tree decomposition $(T, \mathcal{X})$ of a graph is a *star-decomposition* if for each $t \in V(T)$, $X_t \subseteq N[v]$ for some $v \in X_t$. That is, star-decompositions are similar to decompositions of breadth one, but the dominator of each bag has to belong to the bag itself. Lemma 1 shows that both definitions are actually equivalent.

Lemma 1. *For any graph G with $tb(G) \leq 1$, every reduced tree decomposition of G of breadth one is a star-decomposition.*

Proof. Let $(T, \mathcal{X})$ be any reduced tree decomposition of G of breadth one. We will prove it is a star-decomposition. To prove it, let $X_t \in \mathcal{X}$ be arbitrary and let $v \in V$ be such that $\max_{w \in X_t} dist_G(v, w) = 1$, which exists because X_t has radius one. We now show that $v \in X_t$. Indeed, since the subtree T_v and the subtrees $T_w, w \in X_t$, pairwise intersect, then it comes by the Helly Property that $T_v \cap \left(\bigcap_{w \in X_t} T_w\right) \neq \emptyset$ *i.e.*, there is some bag containing $\{v\} \cup X_t$. As a result, we have that $v \in X_t$ because $(T, \mathcal{X})$ is a reduced tree decomposition. The latter implies that $(T, \mathcal{X})$ is a star-decomposition because X_t is arbitrary. □

We then show the main result of this section.

Theorem 1. *Deciding whether a graph has treebreadth one is NP-complete.*

In order to prove Theorem 1, we reduce the following particular instance of Chordal Sandwich (proved to be NP-hard in [9]) to our problem. In [33], the author also proposed a reduction from Chordal Sandwich in order to prove that computing treelength is NP-hard. However, we will need different gadgets than in [33], and we will need different arguments to prove correctness of the reduction.

Problem 1 (Chordal Sandwich with $\overline{nK_2}$).

Input: *graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ such that $E_1 \subseteq E_2$, $|V|$ is even and the complementary $\overline{G}_2$ of G_2 induces a perfect matching.*

Question: *Is there a chordal graph $H = (V, E)$ such that $E_1 \subseteq E \subseteq E_2$?*

Perhaps surprisingly, the restriction on the structure of $\overline{G_2}$ is a key element in our reduction. Indeed, we will need the following technical lemma in what follows.

Lemma 2. *Let $G_1 = (V, E_1)$, $G_2 = (V, E_2)$ such that $E_1 \subseteq E_2$ and $\overline{G_2}$ is a perfect matching. Suppose that $\langle G_1, G_2 \rangle$ is a yes-instance of* Chordal Sandwich with $\overline{nK_2}$.

There exists a tree decomposition $(T, \mathcal{X})$ of G_1 with $|\mathcal{X}| = |V|/2 + 1$ bags such that for every $\{u, v\} \notin E_2$, $T_u \cap T_v = \emptyset$ and there are two adjacent bags $B_u \in T_u$ and $B_v \in T_v$ such that $B_u \setminus u = B_v \setminus v$.

Proof. Let $H = (V, E)$ be any chordal graph such that $E_1 \subseteq E \subseteq E_2$ (that exists because $\langle G_1, G_2 \rangle$ is a yes-instance of Chordal Sandwich with $\overline{nK_2}$ by the hypothesis) and the number $|E|$ of edges is maximized. We will prove that any clique-tree $(T, \mathcal{X})$ of H satisfies the above properties (given in the statement of the lemma). Since it it well known that any clique-tree of H (as chordal supergraph of G_1) is a tree decomposition of G_1, this will prove the lemma. To prove it, let $\{u,v\} \notin E_2$ be arbitrary. Observe that $T_u \cap T_v = \emptyset$ (else, $\{u,v\} \in E$, that would contradict that $E \subseteq E_2$).

Furthermore, let $B_u \in T_u$ minimize the distance in T to the subtree T_v, let B be the unique bag that is adjacent to B_u on a shortest-path between B_u and T_v in T. Note that $B \notin T_u$ by the minimality of $dist_T(B_u, T_v)$, however B may belong to T_v. Removing the edge $\{B_u, B\}$ in T yields two subtrees T_1, T_2 with $T_u \subseteq T_1$ and $T_v \subseteq T_2$. In addition, we have that for every $x \in V \setminus u$ such that $T_x \cap T_1 \neq \emptyset$, $\{u,x\} \in E_2$ since $x \neq v$ and $\overline{G_2}$ is a perfect matching by the hypothesis. Similarly, we have that for every $y \in V \setminus v$ such that $T_y \cap T_2 \neq \emptyset$, $\{v,y\} \in E_2$. Therefore, by maximality of the number $|E|$ of edges, it follows that $T_1 = T_u$ and $T_2 = T_v$, and so, $T_u \cup T_v = T$. In particular, $B = B_v \in T_v$.

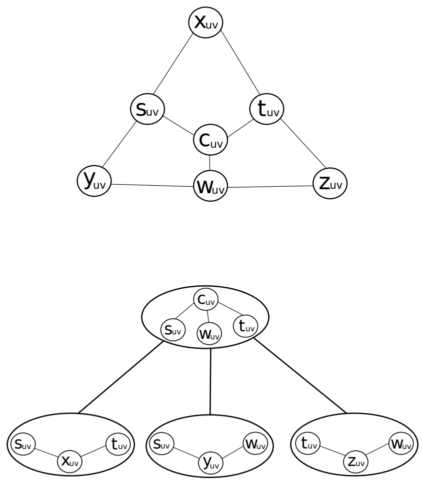
Then, let us prove that $B_u \setminus u = B_v \setminus v$. Indeed, assume for the sake of contradiction that $B_u \setminus u \neq B_v \setminus v$. In particular, $(B_u \setminus B_v) \setminus u \neq \emptyset$ or $(B_v \setminus B_u) \setminus v \neq \emptyset$. Suppose w.l.o.g. that $(B_u \setminus B_v) \setminus u \neq \emptyset$. Let $H' = (V, E')$ be obtained from H by adding an edge between vertex v and every vertex of $(B_u \setminus B_v) \setminus u$. By construction $|E'| > |E|$. Furthermore, H' is chordal since a clique-tree of H' can be obtained from $(T, \mathcal{X})$ by adding a new bag $(B_u \setminus u) \cup \{v\}$ between B_u and B_v. However, for every $x \in (B_u \setminus B_v) \setminus u$ we have that $\{x,v\} \in E_2$ since $x \neq u$ and $\overline{G_2}$ is a perfect matching by the hypothesis. As a result, $E' \subseteq E_2$, thus contradicting the maximality of the number $|E|$ of edges in H.

Overall, the above implies that there is a one-to-one mapping between edges in T and nonedges of G_2. Since $\overline{G_2}$ is a perfect matching, there are $|V|/2$ edges in T, and so, $|V|/2 + 1$ bags. □

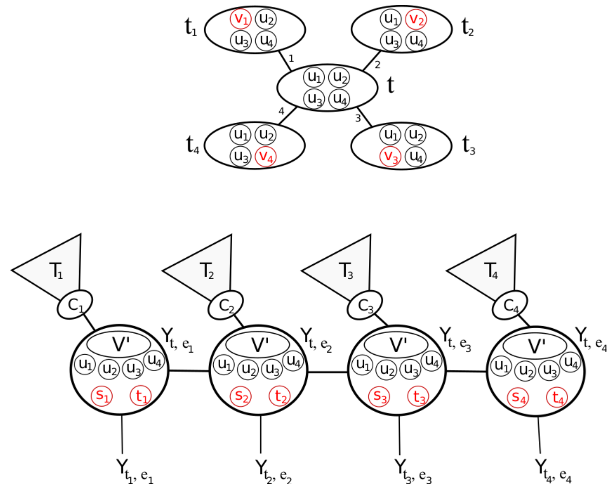
Proof of Theorem 1. The problem is in NP. To prove the NP-hardness, let $\langle G_1, G_2 \rangle$ be any instance of Chordal Sandwich with $\overline{nK_2}$. Let G' be the graph constructed from G_1 as follows. First, a clique V' of $2n = |V|$ vertices is added to G_1. Vertices $v \in V$ are in one-to-one correspondance with vertices $v' \in V'$. Then, for every $\{u,v\} \notin E_2$, u and v are respectively made adjacent to all vertices in $V' \setminus v'$ and $V' \setminus u'$. Finally, we add a copy of the gadget F_{uv}, depicted in Figure 1a, and the vertices s_{uv} and t_{uv} are made adjacent to the four vertices u, v, u', v'.

Roughly, our gadgets ensure that a non-edge of G_2 *cannot* be contained in any bag of a star-decomposition. For that, for every $\{u,v\} \notin E(G_2)$, we include u and v in a cycle of length four, which is not dominated. Part of our gadgets are inspired from those of Lokshtanov for treelength [33] in order to force two vertices to be in a same bag in any tree decomposition of length k. In the same way, we force the two other ends of the cycle containing u and v to be in a same bag in any star-decomposition. Doing so, we can use the Helly property in order to prove that u and v cannot be contained in a common bag (otherwise, the full cycle should be contained in a bag, that could not be dominated). At the same time, we keep in the clique V' one representative per vertex of G_1. For the yes-instances, we will prove the existence of a star-decomposition whose all dominating vertices are in V' (except for some bags that only contain vertices from our gadgets). Roughly, our star-decomposition mimics the tree decomposition with specific properties given by Lemma 2. We recall that the edges of this decomposition are in one-to-one correspondence with the non-edges $\{u,v\}$ of G_2. In our star-decomposition, there are two adjacent bags whose dominating vertices are, respectively, the representatives of u and v in V'. Our proof takes advantage of the fact that V' is a clique: indeed, it allows us to put this subset in *all* the bags of our star-decomposition (again, except for some bags that only contain vertices from our gadgets).

We will prove $tb(G') = 1$ if and only if $\langle G_1, G_2 \rangle$ is a yes-instance of Chordal Sandwich with



(a) Gadget F_{uv} (top) with a star-decomposition of F_{uv} (bottom).



(b) A subtree of the star-decomposition of G' (bottom) obtained from an internal bag with degree four of $(T, \mathcal{X})$ (top). Subtrees T_i are star-decompositions of the gadgets $F_{u_i v_i}$.

Figure 1: The reduction for Theorem 1.

$\overline{nK_2}$.

In one direction, assume $tb(G') = 1$, let $(T, \mathcal{X})$ be a star-decomposition of G' (which exists by Lemma 1). We prove that the triangulation of G_1 obtained from this star-decomposition is the desired chordal sandwich. Let $H = (V, \{\{u, v\} \mid T_u \cap T_v \neq \emptyset\})$. H is a chordal graph such that $E_1 \subseteq E(H)$. To prove that $\langle G_1, G_2 \rangle$ is a yes-instance of Chordal Sandwich with $\overline{nK_2}$, it suffices to prove that $T_u \cap T_v = \emptyset$ for every $\{u, v\} \notin E_2$. We claim that it is implied by $T_{s_{uv}} \cap T_{t_{uv}} \neq \emptyset$. Indeed, assume $T_{s_{uv}} \cap T_{t_{uv}} \neq \emptyset$ and $T_u \cap T_v \neq \emptyset$. Since $s_{uv}, t_{uv} \in N(u) \cap N(v)$, $T_u, T_v, T_{s_{uv}}, T_{t_{uv}}$ pairwise intersect, there is a bag with u, v, s_{uv}, t_{uv} by the Helly property. The latter contradicts that $(T, \mathcal{X})$ is a star-decomposition because no vertex dominates the four vertices. Hence the claim is proved. So, let us prove that $T_{s_{uv}} \cap T_{t_{uv}} \neq \emptyset$. By contradiction, if $T_{s_{uv}} \cap T_{t_{uv}} = \emptyset$ then every bag B onto the path between $T_{s_{uv}}$ and $T_{t_{uv}}$ must contain c_{uv}, x_{uv}. Since $N[c_{uv}] \cap N[x_{uv}] = \{s_{uv}, t_{uv}\}$ and $(T, \mathcal{X})$ is a star-decomposition, it implies either $s_{uv} \in B$ and $B \subseteq N[s_{uv}]$ or $t_{uv} \in B$ and $B \subseteq N[t_{uv}]$. So, there are two adjacent bags $B_s \in T_{s_{uv}}, B_t \in T_{t_{uv}}$ such that $B_s \subseteq N[s_{uv}]$ and $B_t \subseteq N[t_{uv}]$. In particular, $B_s \cap B_t$ must intersect the path (y_{uv}, w_{uv}, z_{uv}) because $y_{uv} \in N(s_{uv})$ and $z_{uv} \in N(t_{uv})$. However, $N[s_{uv}] \cap N[t_{uv}] \cap \{y_{uv}, w_{uv}, z_{uv}\} = \emptyset$, that is a contradiction. As a result, $T_{s_{uv}} \cap T_{t_{uv}} \neq \emptyset$ and so, $T_u \cap T_v = \emptyset$ for any $\{u, v\} \notin E_2$.

Conversely, assume that $\langle G_1, G_2 \rangle$ is a yes-instance of Chordal Sandwich with $\overline{nK_2}$. Since $\overline{G_2}$ is a perfect matching by the hypothesis, let $(T, \mathcal{X})$ be as stated in Lemma 2. We will modify $(T, \mathcal{X})$ in order to obtain a star-decomposition of G'. To do so, we will use the fact that there are $|V|/2 = n$ edges in $E(T)$ and the properties stated by Lemma 2. Indeed, this implies that there is a one-to-one mapping $\alpha : E(T) \to E(\overline{G}_2)$ between the edges of T and the non-edges of G_2. Precisely, for any edge $e = \{t, s\} \in E(T)$, let $\alpha(e) = \{u, v\} \in E(\overline{G}_2)$ be the non-edge of G_2 such that $u \in X_t, v \in X_s$ and $X_t \setminus u = X_s \setminus v$.

Intuitively, the star-decomposition $(T', \mathcal{X}')$ of G' is obtained as follows. For any $t \in V(T)$ with incident edges $e_1, \cdots, e_d$, we first replace X_t by a path decomposition $(Y_{t,e_1}, \cdots, Y_{t,e_d})$. Then, for any edge $e = \{t, s\} \in E(T)$, an edge is added between $Y_{t,e}$ and $Y_{s,e}$. Finally, the center-bag of some star-decomposition of the gadget $F_{\alpha(e)}$ is made adjacent to $Y_{t,e}$ (see Figure 1b for an illustration).

More formally, let $t \in V(T)$ and $e \in E(T)$ incident to t, and let $\{u, v\} = \alpha(e)$. Let $Y_{t,e} = V' \cup X_t \cup \{s_{uv}, t_{uv}\}$ (note that $Y_{t,e}$ is dominated by $u' \in V'$). Let $e_1, \cdots, e_d$ be the edges incident to t in T, in any order. For $1 \leq i < d$, add an edge between Y_{t,e_i} and $Y_{t,e_{i+1}}$. For any edge $e = \{t, s\} \in E(T)$, add an edge between $Y_{t,e}$ and $Y_{s,e}$. Finally, add the star-decomposition $(T^e, \mathcal{X}^e)$ for the gadget $F_{\alpha(e)}$ as depicted in Figure 1a and add an edge between its center and $Y_{t,e}$.

The resulting $(T', \mathcal{X}')$ is a star-decomposition of G', hence $tb(G') = 1$. $\square$

We next show that computing the treebreadth is polynomially equivalent to the recognition of graphs with treebreadth one.

Lemma 3. *For every graph G, for every positive integer r, there exists a graph G'_r computable in polynomial time such that $tb(G) \leq r$ if and only if $tb(G'_r) \leq 1$.*

Proof. Let G have vertices $v_1, v_2, \ldots, v_n$, and let $r > 0$. The graph G'_r is obtained from G by adding a clique $U = \{u_1, u_2, \ldots, u_n\}$ so that for every $1 \leq i \leq n$, u_i is adjacent to all vertices in $B_G(v_i, r) = \{x \in V(G) \mid dist_G(v_i, x) \leq r\}$.

If $tb(G) \leq r$ then we claim that given a tree decomposition $(T, \mathcal{X})$ of G with breadth at most r, one obtains a star-decomposition of G'_r by adding the clique U in every bag in $\mathcal{X}$. Indeed, for every bag $X_t \in \mathcal{X}$, by the hypothesis there is $v_i \in V(G)$ such that $\max_{x \in X_t} dist_G(v_i, x) \leq r$, hence $X_t \cup U \subseteq N_{G'_r}[u_i]$. Conversely, if $tb(G'_r) \leq 1$ then we claim that given a star-decomposition $(T', \mathcal{X}')$ of G'_r, one obtains a tree decomposition of G with breadth at most r by removing every vertex of the clique U from every bag in $\mathcal{X}'$. Indeed, for every bag $X'_t \in \mathcal{X}'$, by the hypothesis there is $y \in X'_t$ such that $X'_t \subseteq N_{G'_r}[y]$. Furthermore, $y \in \{u_i, v_i\}$ for some $1 \leq i \leq n$, and so, since $N_{G'_r}[v_i] \subseteq N_{G'_r}[u_i]$ by construction, $X'_t \setminus U \subseteq N_{G'_r}(u_i) \setminus U = \{x \in V(G) \mid dist_G(v_i, x) \leq r\}$. $\square$

Lemma 4. *For every graph G, for every positive integer r, there exists a graph G' computable in polynomial time such that $tb(G) \leq 1$ if and only if $tb(G') \leq r$.*

Proof. For every $\{u, v\} \in E(G)$, let F^r_{uv} be obtained from F_{uv} in Figure 1a by adding an edge $\{s_{uv}, t_{uv}\}$ then subdividing each edge $r-1$ times. The graph G' is obtained from G by substituting every edge $\{u, v\} \in E(G)$ with a distinct copy of F^r_{uv} then identifying u, v with s_{uv}, t_{uv}.

If $tb(G) \leq 1$ then let us modify a star-decomposition $(T, \mathcal{X})$ of G in a tree decomposition $(T', \mathcal{X}')$ of G' of breadth at most r. Clearly, every bag in $\mathcal{X}$ has radius at most r in G'. Furthermore, let $(T^{uv}, \mathcal{X}^{uv})$ be the star-decomposition of F_{uv} in Figure 1a, with three leaf-bags and one central bag. It can be modified in a tree decomposition of F^r_{uv} by `i)` adding in each bag containing both end-vertices of an edge in F_{uv} the $r-1$ vertices in F^r_{uv} that result from its subdivision, and `ii)` adding a new leaf-bag with $\{u, v\}$ and the $r-1$ vertices that result from its subdivision. Finally, let $(T', \mathcal{X}')$ be obtained from $(T, \mathcal{X})$ by adding an edge between some bag in $T_u \cap T_v$ and the central bag of T^{uv} for every $\{u, v\} \in E(G)$. Since $(T', \mathcal{X}')$ has breadth r, $tb(G') \leq r$.

Conversely, if $tb(G') \leq r$ then we start from a tree decomposition $(T', \mathcal{X}')$ of G' of breadth at most r that maximizes the number of pairs $u, v \in V(G)$ such that: $\{u, v\} \in E(G)$ and $T'_u \cap T'_v \neq \emptyset$. We claim that one obtains a tree decomposition of G of breadth one by removing every vertex of $V(G') \setminus V(G)$ from the bags in $\mathcal{X}'$. Before proving the claim, we first make the following useful observation: since every vertex $x \in V(G') \setminus V(G)$ is contained in a gadget $F^r_{uv} \setminus \{u, v\}$, for some edge $\{u, v\} \in E(G)$, such a vertex is at distance at least $r+1$ from any vertex of $V(G) \setminus \{u, v\}$. In particular, for any bag $X'_t \in \mathcal{X}'$ such that $X'_t \subseteq B_{G'}(x, r)$, we have that $X'_t \cap V(G) \subseteq \{u, v\}$ has a radius at most one in G. Furthermore, since $dist_{G'}(v,w) = r \cdot dist_G(v,w)$ for every $v,w \in V(G)$, we also have that for every $v \in V(G)$, for any bag $X'_t \in \mathcal{X}'$ such that $X'_t \subseteq B_{G'}(v,r)$, $X'_t \cap V(G) \subseteq N_G[v]$ has a radius at most one in G. Therefore, in order to prove the claim it suffices to prove that $u = s_{uv}$ and $v = t_{uv}$ are in a common bag of $\mathcal{X}'$ for every $\{u,v\} \in E(G)$. In what follows, we explain how the latter can be proved by elaborating on the same arguments as for Theorem 1.

More precisely, consider the $s_{uv}t_{uv}$-paths P_x, P_c, P_w in F^r_{uv} that are obtained, respectively, from the subdivision of the paths (s_{uv}, x_{uv}, t_{uv}), (s_{uv}, c_{uv}, t_{uv}) and $(s_{uv}, y_{uv}, w_{uv}, z_{uv}, t_{uv})$ in F_{uv}. We suppose for the sake of contradiction that $T'_u \cap T'_v = \emptyset$. By the properties of tree decompositions, any bag B onto the shortest $T'_uT'_v$-path in T' must intersect the three of $P_x \setminus \{u,v\}, P_c \setminus \{u,v\}, P_w \setminus \{u,v\}$. It implies $B \subseteq B_{G'}(b,r)$ for some $b \in B_{G'}(u,r-1) \cup B_{G'}(v,r-1)$. We prove as a subclaim that there are two such bags B_u, B_v and two vertices $b_u \in B_{G'}(u,r-1) \setminus B_{G'}(v,r)$, $b_v \in B_{G'}(v,r-1) \setminus B_{G'}(u,r)$ such that $B_u \subseteq B_{G'}(b_u,r)$, $B_v \subseteq B_{G'}(b_v,r)$. Indeed, otherwise one of u or v could be put in all the bags onto the shortest $T'_uT'_v$-path in T', thereby contradicting the maximality of T'. So, the subclaim is proved. However, consider now vertex w_{uv}. Since $dist_{G'}(u,w_{uv}) = dist_{G'}(v,w_{uv}) = 2r > r + r - 1$, we deduce that $w_{uv} \notin B$ for any bag B onto the shortest $T'_uT'_v$-path in T'. Let us write $P_w = (u, Q_u, w_{uv}, Q_v, v)$. By the properties of tree decompositions, there are two cases. Either for every B onto the shortest $T'_uT'_v$-path in T' we have $B \cap Q_u \neq \emptyset$, or for any such bag B we have $B \cap Q_v \neq \emptyset$. In particular, either $B_v \cap Q_u \neq \emptyset$ or $B_u \cap Q_v \neq \emptyset$. In both cases we derive a contradiction since $B_{G'}(b_u,r) \cap Q_v = B_{G'}(b_v,r) \cap Q_u = \emptyset$. So, the claim is proved, and we obtain that $tb(G) \leq 1$. □

From Lemmas 3, 4 and Theorem 1, it follows that:

Theorem 2. *For any fixed $k \geq 1$, deciding whether a graph G has treebreadth at most k is NP-complete.*

2.2 Pathlength and pathbreadth

To conclude this section, we consider pathlength and pathbreadth.

Theorem 3. *The following two problems are NP-complete:*

- *Deciding whether a graph G has pathlength at most 2 (even if G has diameter at most 3);*
- *Deciding whether a graph G has pathbreadth at most 1 (even if G has radius at most 2).*

We use the Betweenness problem, defined below, in order to prove Theorem 3. Betweenness, sometimes called Total Ordering, is NP-complete [35]. In [28], it was used to show that the Interval Sandwich problem is NP-complete. By Theorem 3, Interval Sandwich remains NP-complete even if the second graph is the *square* of the first one (where the square G^2 of any graph G is obtained from G by adding an edge between every two distinct vertices that are at distance at most 2 in G). Indeed, a graph G has pathlength at most 2 if and only if there is an Interval Sandwich between G and G^2; we refer to [33] for the proof of a similar equivalence between treelength and Chordal Sandwich.

Problem 2 (Betweenness).

Input: *a set $\mathcal{S}$ of n elements, a set $\mathcal{T}$ of m ordered triples of elements in $\mathcal{S}$.*

Question: *Is there a total ordering of $\mathcal{S}$ such that for every triple $t = (s_i, s_j, s_k) \in \mathcal{T}$, either $s_i < s_j < s_k$ or $s_k < s_j < s_i$?*

We remark that, by the above definition, reversing any triple of $\mathcal{S}$ does not change the answer. Given any instance $(\mathcal{S}, \mathcal{T})$ of Betweenness, we construct from $\mathcal{S}$ and $\mathcal{T}$ a graph $G_{\mathcal{S},\mathcal{T}}$ as defined below. We will then prove that $pl(G_{\mathcal{S},\mathcal{T}}) \leq 2$ (resp. $pb(G_{\mathcal{S},\mathcal{T}}) \leq 1$) if and only if $(\mathcal{S}, \mathcal{T})$ is a yes-instance of Betweenness.

Definition 4. Let $\mathcal{S}$ be a set of n elements, let $\mathcal{T}$ be a set of m ordered triples of elements in $\mathcal{S}$. The graph $G_{\mathcal{S},\mathcal{T}}$ is constructed as follows:

- For every element $s_i \in \mathcal{S}$, $1 \leq i \leq n$, there are two adjacent vertices u_i, v_i in $G_{\mathcal{S},\mathcal{T}}$. The vertices u_i are pairwise adjacent *i.e.*, the set $U = \{u_i \mid 1 \leq i \leq n\}$ is a clique.
- For every triple $t = (s_i, s_j, s_k) \in \mathcal{T}$, let us add in $G_{\mathcal{S},\mathcal{T}}$ the v_iv_j-path (v_i, a_t, b_t, v_j) of length 3, and the v_jv_k-path (v_j, c_t, d_t, v_k) of length 3.
- Finally, for every triple $t = (s_i, s_j, s_k) \in \mathcal{T}$ let us make adjacent a_t, b_t with every u_l such that $l \neq k$, similarly let us make adjacent c_t, b_t with every u_l such that $l \neq i$.

It can be noticed from Definition 4 that for any $1 \leq i \leq n$, the vertex u_i is adjacent to every vertex, except: the vertices v_j such that $j \neq i$; the vertices a_t, b_t such that s_i is the last element of triple t; and the vertices c_t, b_t such that s_i is the first element of triple t. We refer to Figure 2 for an illustration (see also Figure 3). Observe that $G_{\mathcal{S},\mathcal{T}}$ has diameter 3 because the clique U dominates $G_{\mathcal{S},\mathcal{T}}$. Therefore $pl(G_{\mathcal{S},\mathcal{T}}) \leq 3$ and we will show that it is hard to distinguish graphs with pathlength two from graphs with pathlength three. Similarly, since the clique U dominates $G_{\mathcal{S},\mathcal{T}}$ we have $pb(G_{\mathcal{S},\mathcal{T}}) \leq 2$ and we will show that it is hard to distinguish graphs with pathbreadth one from graphs with pathbreadth two.

The correctness of our reduction is proved in the following two Lemmas 5 and 6.

Lemma 5. *Let $\mathcal{S}$ be a set of n elements, let $\mathcal{T}$ be a set of m ordered triples of elements in $\mathcal{S}$. If $(\mathcal{S}, \mathcal{T})$ is a yes-instance of* Betweenness *then $pb(G_{\mathcal{S},\mathcal{T}}) \leq 1$ and $pl(G_{\mathcal{S},\mathcal{T}}) \leq 2$, where $G_{\mathcal{S},\mathcal{T}}$ is the graph of Definition 4.*

Proof. Since $pl(G_{\mathcal{S},\mathcal{T}}) \leq 2 \cdot pb(G_{\mathcal{S},\mathcal{T}})$ we only need to prove that $pb(G_{\mathcal{S},\mathcal{T}}) \leq 1$. For convenience, let us reorder the elements of $\mathcal{S}$ so that for every triple $(s_i, s_j, s_k) \in \mathcal{T}$ either $i < j < k$ or $k < j < i$. It is possible to do that because by the hypothesis $(\mathcal{S}, \mathcal{T})$ is a yes-instance of Betweenness. If furthermore $k < j < i$, let us also replace (s_i, s_j, s_k) with the inverse triple (s_k, s_j, s_i). This way, we have a total ordering of $\mathcal{S}$ such that $s_i < s_j < s_k$ for every triple $(s_i, s_j, s_k) \in \mathcal{T}$. Then, let us construct a path decomposition $(P, \mathcal{X})$ with n bags, denoted $X_1, X_2, \ldots, X_n$, as follows:

- For every $1 \leq i \leq n$, we add the clique U and the vertex v_i in X_i.
- For every $t = (s_i, s_j, s_k) \in \mathcal{T}$, we add both a_t, b_t in the bags X_l with $i \leq l \leq j$. Similarly we add both c_t, d_t in the bags X_l with $j \leq l \leq k$.

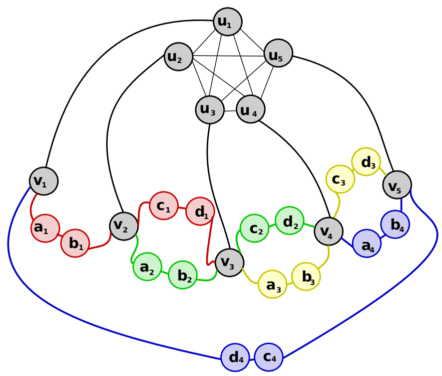


Figure 2: The graph $G_{\mathcal{S},\mathcal{T}}$ for $\mathcal{S} = [|1,5|]$ and $\mathcal{T} = \{(i, i+1, i+2) \mid 1 \leq i \leq 4\}$. For ease of reading, the adjacency relations between the vertices u_i and a_t, b_t, c_t, d_t are not drawn.

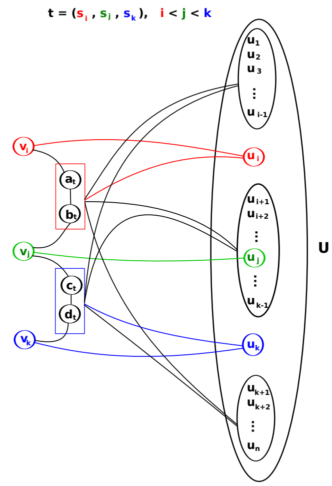


Figure 3: Adjacency relations in $G_{\mathcal{S},\mathcal{T}}$ for one given triple $t = (s_i, s_j, s_k)$.

By construction, the clique U is contained in any bag of P. Furthermore for every triple $t = (s_i, s_j, s_k) \in \mathcal{T}$ we have: $a_t, b_t, v_i \in X_i$; $a_t, b_t, c_t, d_t, v_j \in X_j$; and $c_t, d_t, v_k \in X_k$. Therefore, every vertex and every edge of $G_{\mathcal{S},\mathcal{T}}$ is contained in at least one bag. Moreover, by construction the bags containing any vertex are consecutive. Hence, $(P, \mathcal{X})$ is indeed a path decomposition of $G_{\mathcal{S},\mathcal{T}}$.

We claim that for every i, $X_i \subseteq N[u_i]$, that will prove the lemma. Indeed if it were not the case for some i then by Definition 4 there should exist $t \in \mathcal{T}, j, k$ such that: either $t = (s_i, s_j, s_k) \in \mathcal{T}$ and $c_t, d_t \in X_i$; or $t = (s_k, s_j, s_i) \in \mathcal{T}$ and $a_t, b_t \in X_i$. But then by construction either a_t, b_t are only contained in the bags X_l for every $k \leq l \leq j$, or c_t, d_t are only contained in the bags X_l for every $j \leq l \leq k$, thereby contradicting the fact that either $a_t, b_t \in X_i$ or $c_t, d_t \in X_i$. □

Lemma 6. *Let $\mathcal{S}$ be a set of n elements, let $\mathcal{T}$ be a set of m ordered triples of elements in $\mathcal{S}$. If $pb(G_{\mathcal{S},\mathcal{T}}) \leq 1$ or $pl(G_{\mathcal{S},\mathcal{T}}) \leq 2$ then $(\mathcal{S}, \mathcal{T})$ is a yes-instance of* Betweenness*, where $G_{\mathcal{S},\mathcal{T}}$ is the graph of Definition 4.*

Proof. Since $pl(G_{\mathcal{S},\mathcal{T}}) \leq 2 \cdot pb(G_{\mathcal{S},\mathcal{T}})$ then we only need to consider the case when $pl(G_{\mathcal{S},\mathcal{T}}) \leq 2$. Let $(P, \mathcal{X})$ be a path decomposition of length two, that exists by the hypothesis. Since the vertices v_i are pairwise at distance 3 then the subpaths P_{v_i} that are induced by the bags containing vertex v_i are pairwise disjoint. Therefore, starting from an arbitrary endpoint of P and considering each vertex v_i in the order that it appears in the path decomposition, this defines a total ordering over $\mathcal{S}$. Let us reorder the set $\mathcal{S}$ so that vertex v_i is the i^{th} vertex to appear in the path decomposition. We claim that for every triple $t = (s_i, s_j, s_k) \in \mathcal{T}$, either $i < j < k$ or $k < j < i$, that will prove the lemma.

By way of contradiction, let $t = (s_i, s_j, s_k) \in \mathcal{T}$ such that either $j < \min\{i, k\}$ or $j > \max\{i, k\}$. We only need to consider the case when $j < i < k$ because all the other cases are symmetrical to this one. In a such case by construction the unique path in P between P_{v_j} and P_{v_k} contains P_{v_i}. Let $B \in P_{v_i}$, by the properties of a tree decomposition it is a $v_j v_k$-separator, so it must contain one of c_t, d_t. However, vertex $v_i \in B$ is at distance 3 from both vertices c_t, d_t, thereby contradicting the fact that $(P, \mathcal{X})$ has length 2. □

We are now able to prove Theorem 3.

Proof of Theorem 3. We prove the two statements of the theorem separately.

First, we claim that deciding whether a graph has pathlength at most k is NP-complete, even if $k = 2$. Indeed, in order to prove that a graph G satisfies $pl(G) \leq k$, it suffices to give as a certificate a tree decomposition of G with length at most k (since the all-pairs-shortest-paths in G can be computed in polynomial-time). Therefore, the problem is in NP. Then, given an instance $(\mathcal{S}, \mathcal{T})$ of Betweenness, let $G_{\mathcal{S},\mathcal{T}}$ be the graph of Definition 4. We prove that $pl(G_{\mathcal{S},\mathcal{T}}) \leq 2$ if and only if the pair $(\mathcal{S}, \mathcal{T})$ is a yes-instance of Betweenness. This will prove the NP-hardness because our reduction is polynomial and Betweenness is NP-complete. In one direction, if $(\mathcal{S}, \mathcal{T})$ is a yes-instance then by Lemma 5 $pl(G_{\mathcal{S},\mathcal{T}}) \leq 2$. Conversely, if $pl(G_{\mathcal{S},\mathcal{T}}) \leq 2$ then $(\mathcal{S}, \mathcal{T})$ is a yes-instance by Lemma 6, that proves the other direction. So, the first claim is proved.

Similarly, we claim that deciding whether a graph has pathbreadth at most k is NP-complete, even if $k = 1$. Indeed, in order to prove that a graph G satisfies $pb(G) \leq k$, it suffices to give as a certificate a tree decomposition of G with breadth at most k. Therefore, the problem is in NP. We prove that $pb(G_{\mathcal{S},\mathcal{T}}) \leq 1$ if and only if the pair $(\mathcal{S}, \mathcal{T})$ is a yes-instance of Betweenness. Again, this will prove the NP-hardness because our reduction is polynomial and Betweenness is NP-complete. In one direction, if $(\mathcal{S}, \mathcal{T})$ is a yes-instance then by Lemma 5 $pb(G_{\mathcal{S},\mathcal{T}}) \leq 1$. Conversely, if $pb(G_{\mathcal{S},\mathcal{T}}) \leq 1$ then $(\mathcal{S}, \mathcal{T})$ is a yes-instance by Lemma 6, that proves the other direction. So, this claim is also proved. □

We let open whether, for every $k \geq 1$, the recognition of graphs with pathbreadth at most k, resp. with pathlength at most $k + 1$, is NP-complete.

3 Graphs with treebreadth one: some polynomial cases

In this section, we investigate further the class of graphs with treebreadth one. It strictly contains chordal graphs and dually chordal graphs, well-studied graph classes in algorithmic graph theory [13]. We first show some useful lemmas that somehow state that we can restrict our study on graphs without clique-separator. Then, we show that the problem of recognizing graphs with treebreadth one can be solved in polynomial time in the class of bipartite graphs and in the class of $K_{3,3}$-minor-free graphs.

Let $G = (V, E)$ be a connected graph. Recall that a set $S \subset V$ is a *separator* if $G \setminus S$ is disconnected. It is called a *clique-separator* if S induces a complete graph. A *full component* for S is any connected component C of $G \setminus S$ such that $N(C) = S$. If C is a full component for S then we call the induced subgraph $G[C \cup S]$ a block. Finally, S is a *minimal separator* if there exist at least two full components for S.

Our objective is to prove that if a graph G has treebreadth one then so do all its blocks. We stress that in general, treebreadth is not closed under taking induced subgraphs. Therefore, it comes a bit as a surprise that graphs of treebreadth one are closed under taking blocks. In fact, we will prove a slightly more general result:

Lemma 7. *Let $G = (V, E)$, S be a separator and W be the union of some connected components of $G \setminus S$. If $tb(G) = 1$ and W contains a full component for S, then $tb(G[W \cup S]) = 1$.*

Proof. Let $(T, \mathcal{X})$ be a star-decomposition of G. We remove vertices in $V \setminus (W \cup S)$ from bags in $\mathcal{X}$, that yields a tree decomposition $(T, \mathcal{X}')$ of $G[W \cup S]$. We will prove that $(T, \mathcal{X}')$ has breadth one (but is not necessarily a star-decomposition). Indeed, let $X'_t \in \mathcal{X}'$. By construction, $X'_t \subseteq X_t$ with $X_t \in \mathcal{X}$. Let $v \in X_t$ satisfy $X_t \subseteq N_G[v]$. If $v \in X'_t$, then we are done. Else, since for all

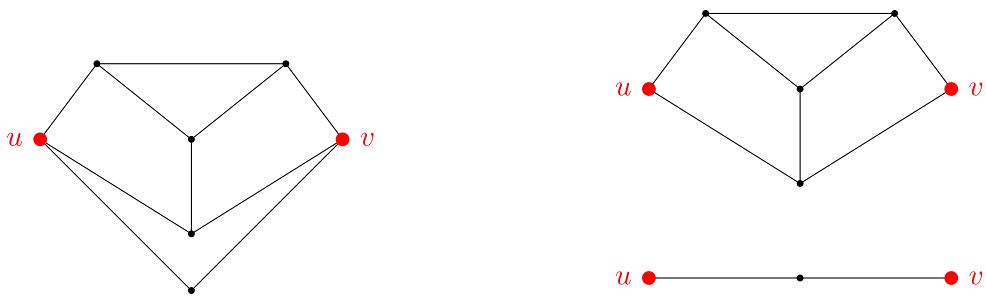


Figure 4: The 2-separator $\{u, v\}$ disconnects the graph $G = (V, E)$ (left) in two blocks with treebreadth one (right). However, $tb(G) = 2$. Indeed, on one hand let w be the unique vertex of $V \setminus (N_G[u] \cup N_G[v])$. Then, the tree decomposition with the two bags $\{u, v\} \cup (N_G(u) \cap N_G(v))$ and $\{u, v\} \cup N_G[w]$ (of size 4 and 6, respectively) has clearly breadth two. On the other hand, suppose by contradiction that there exists a star-decomposition $(T, \mathcal{X})$ for G. We observe that there is no vertex whose closed neighbourhood intersects the internal nodes of all the uv-paths. Therefore, we must have $T_u \cap T_v \neq \emptyset$ (otherwise, by the properties of tree decompositions u and v are disconnected by the intersection between two adjacent bags in T, but by our previous observation such a uv-separator could not be dominated). Now, consider the unique uv-path (u, x, y, v) of length three. Since it is not dominated, either $T_u \cap T_y = \emptyset$ or $T_v \cap T_x = \emptyset$. However, there is no vertex whose closed neighbourhood intersects the internal nodes of all the uy-paths (of all the vx-paths, resp.), that is a contradiction.

$x \notin S \cup W, N(x) \cap (S \cup W) \subseteq S$ (because S is a separator by the hypothesis), we must have that $X'_t \subseteq S$. Let $A \subseteq W$ be a full component for S, that exists by the hypothesis, let T_A be induced by the bags intersecting A. Since T_A and the subtrees $T_x, x \in X'_t$ pairwise intersect — because for all $x \in X'_t$, $x \in S$ and so, x has a neighbour in A —, then by the Helly property there is a bag in $\mathcal{X}$ containing X'_t and intersecting A. Furthermore, any $u \in V$ dominating this bag must be either in S or in A, so, in particular there is $u \in A \cup S$ such that $X'_t \subseteq N[u]$. □

The converse of Lemma 7 does not hold in general (see Fig. 4), yet there are interesting cases when it does.

Lemma 8. *Let $G = (V, E)$ with a minimal separator S inducing a complete graph, and let A be a full component. Then, $tb(G) = 1$ if and only if $tb(G[A \cup S]) = 1$ and $tb(G[V \setminus A]) = 1$.*

Proof. By the hypothesis $V \setminus (A \cup S)$ contains a full component because S is a minimal separator. Therefore, if G has treebreadth one, then so do $G[A \cup S]$ and $G[V \setminus A]$ by Lemma 7 (this also follows the fact that treebreadth is contraction-closed [21, 22] and S is a complete graph). Conversely, suppose that we have both $tb(G[A \cup S]) = 1$ and $tb(G[V \setminus A]) = 1$. Let $(T^1, \mathcal{X}^1)$ be a tree decomposition of $G[A \cup S]$ with breadth one, let $(T^2, \mathcal{X}^2)$ be a tree decomposition of $G[V \setminus A]$ with breadth one. It is well known (by the Helly Property) that any clique must be fully contained in some bag of every tree decomposition. Therefore, S is fully contained into some bag of $(T^1, \mathcal{X}^1)$ and it is fully contained into some bag of $(T^2, \mathcal{X}^2)$. Moreover, $(A \cup S) \cap (V \setminus A) = S$, therefore a tree decomposition of G with breadth one can be obtained by adding an edge between some bag of $(T^1, \mathcal{X}^1)$ containing S and some bag of $(T^2, \mathcal{X}^2)$ containing S. □

Recall that computing the clique-minimal-separator decomposition of a graph G takes $\mathcal{O}(nm)$-time, where m (resp., n) denotes the number of edges (resp., of vertices) [6]. By doing so, one replaces a graph G with the maximal subgraphs of G that have no clique-separator, *a.k.a. atoms*. So, in the following we will only consider graphs without a clique-separator, *a.k.a.*, *prime graphs*.

3.1 Bipartite graphs and triangle-free graphs

Bipartite graphs with treebreadth one are an interesting subclass of their own since they contain the convex bipartite graphs (*i.e.*, bipartite graphs whose vertices on one side can be enumerated such that for every vertex v on the other side the vertices adjacent to v are consecutive) and the chordal bipartite graphs (*i.e.*, bipartite graphs with no induced cycle of length at least six). In this section, we present a linear-time algorithm that decides whether a prime bipartite graph has treebreadth one, and computes a corresponding decomposition if any. Since the clique-minimal-separator decomposition of a given bipartite graph can be computed in linear time [5], this proves combined with Lemma 8 that it can be decided in linear time whether a bipartite graph has treebreadth one.

More precisely, we show that prime bipartite graphs with treebreadth one coincide with *tree-convex* bipartite graphs, a generalization of convex bipartite graphs [38]. A bipartite graph is called tree-convex if it admits a tree decomposition where the bags are the close neighbourhoods of any one side of its bipartition. That is, $G = (V_0 \cup V_1, E)$ is tree-convex if there is a tree decomposition $(T, \mathcal{X})$ such that the set of bags $\mathcal{X}$ is precisely either $\{N[v_0] \mid v_0 \in V_0\}$, or $\{N[v_1] \mid v_1 \in V_1\}$. By definition, tree-convex graphs have treebreadth one. The following lemma is a converse of this result.

Lemma 9. *Let $G = (V_0 \cup V_1, E)$ be a prime bipartite graph with treebreadth one. There is $(T, \mathcal{X})$ a star-decomposition of G such that either $\mathcal{X} = \{N[v_0] \mid v_0 \in V_0\}$, or $\mathcal{X} = \{N[v_1] \mid v_1 \in V_1\}$. In particular, G is tree-convex.*

Proof. Let $(T, \mathcal{X})$ be a star-decomposition of G minimizing $|\mathcal{X}|$. W.l.o.g., suppose there is some $v_0 \in V_0$, there is $t \in V(T)$ such that $X_t \subseteq N_G[v_0]$. We claim that for every $t' \in V(T)$, there is $v_0' \in V_0$ such that $X_{t'} \subseteq N_G[v_0']$. By contradiction, let $v_0 \in V_0, v_1 \in V_1$, let $t, t' \in V(T)$ be such that $X_t \subseteq N_G[v_0], X_{t'} \subseteq N_G[v_1]$. By connectivity of the tree T we may assume w.l.o.g. that $\{t, t'\} \in E(T)$. Moreover, $N_G(v_0) \cap N_G(v_1) = \emptyset$ because G is bipartite. Therefore, $X_t \cap X_{t'} \subseteq \{v_0, v_1\}$, and in particular if $X_t \cap X_{t'} = \{v_0, v_1\}$ then v_0, v_1 are adjacent in G (since $X_t \subseteq N_G[v_0], X_{t'} \subseteq N_G[v_1]$). However, by the properties of a tree decomposition this implies that $X_t \cap X_{t'}$ is a clique-separator (either an edge or a single vertex), thus contradicting the fact that G is prime. It follows, as claimed, that for every $t' \in V(T)$, there is $v_0' \in V_0$ such that $X_{t'} \subseteq N_G[v_0']$.

In order to complete the proof of the lemma, let $v_0 \in V_0$ be arbitrary. We claim that there is a unique bag X_t, $t \in V(T)$, containing v_0. Indeed, any such bag X_t must satisfy $X_t \subseteq N_G[v_0]$, hence the subtree T_{v_0} can be contracted into a single bag $\bigcup_{t \in T_{v_0}} X_t$ without violating the property for the tree decomposition to be a star-decomposition. As a result, the uniqueness of the bag X_t follows from the minimality of $|\mathcal{X}|$. Since X_t is unique and $X_t \subseteq N_G[v_0]$, therefore $X_t = N_G[v_0]$ (since all edges must be in some bag) and so, $\mathcal{X} = \{N[v_0] \mid v_0 \in V_0\}$. □

Combining Lemmas 8 and 9, we obtain the following characterization of bipartite graphs with treebreadth one.

Corollary 5. *A bipartite graph has treebreadth one if and only if every of its atoms is tree-convex. It can be decided in linear time.*

Proof. Given a bipartite graph G, we can check whether it has treebreadth one as follows. We compute its atoms, that can be done in linear time [5]. Then, we check whether every atom of G is tree-convex. As shown in [38], it can also be done in linear time by reducing this problem to the recognition of dual hypertrees. Finally, by Lemma 9 we output $tb(G) = 1$ if and only if all the atoms of G are tree-convex. □

In fact we can also decide, for any given *triangle-free* graph G, whether $tb(G) = 1$. Indeed, w.l.o.g. G is prime for clique-decomposition. Then, suppose $tb(G) = 1$ and let $(T, \mathcal{X})$ be a star-decomposition minimizing $|\mathcal{X}|$. For every $\{t, t'\} \in E(T)$ with $X_t \subseteq N_G[v],\ X_{t'} \subseteq N_G[u]$ we claim that $\{u, v\} \notin E(G)$. Indeed, otherwise $X_t \cap X_{t'} \subseteq N_G[u] \cap N_G[v]$ would be a clique-separator (either a cut-vertex or the edge-separator $\{u, v\}$), that follows from the fact that G is triangle-free. Then, all bags containing v, resp. containing u, can only be dominated by v itself, resp. by u itself. By minimality of $|\mathcal{X}|$ we deduce that $X_t = N_G[v],\ X_{t'} = N_G[u]$. Said otherwise, every bag of the tree decomposition is the closed neighbourhood of a vertex, *i.e.*, G has strong treebreadth one. Leitert and Dragan proved that for every graph with strong treebreadth ρ, a tree decomposition of breadth at most ρ can be computed in $\mathcal{O}(n^2m)$-time. As a result, we can decide whether $tb(G) = 1$ within the same amount of time.

3.2 Planar graphs and beyond

In the conference version of this paper [25], we sketched a quadratic-time algorithm to recognize prime planar graphs of treebreadth one (see also [26]). This algorithm was based on intricate reduction rules. Here we propose a conceptually simpler algorithm, that decides in $\mathcal{O}(n^3m)$-time whether a prime $K_{3,3}$-minor-free graph has unit treebreadth[2]. Combined with Lemma 8, this shows that $K_{3,3}$-minor-free graphs of treebreadth one can be recognized in polynomial time.

Theorem 6. *Recognizing $K_{3,3}$-minor-free graphs of treebreadth one can be done in $\mathcal{O}(n^3m)$-time. Moreover, a star-decomposition (if any) can also be computed in $\mathcal{O}(n^3m)$-time.*

Note that planar graphs with treebreadth one contain the Apollonian graphs, that are exactly the chordal maximal planar graphs and have received some attention in the litterature of random networks [2]. In particular, every Apollonian graph has treewidth at most three. More generally, the class of $K_{3,3}$-minor-free graphs of treebreadth one has bounded treewidth [16] (see also Corollary 11). Therefore, our work in this section brings more insights on tree decompositions with small width for $K_{3,3}$-minor-free graphs, and so, in particular for planar graphs. We do not use the bounded treewidth property for proving Theorem 6. However, our algorithm in this section is close in spirit to the algorithmic scheme of Bodlaender and Kloks for computing treewidth [10]. For the latter, a tree decomposition with small, but possibly suboptimal width, is first computed. Then, a tree decomposition of optimal width is computed by dynamic programming on the first (suboptimal) decomposition. In the same way, we first compute a so called “quasi star-decomposition” of the graph G, that is a specific type of tree decomposition of breadth at most four (Definition 10). Then, we perform dynamic programming on this tree decomposition in order to compute a star-decomposition of G (if any) by splitting some bags.

[2]We think that the running-time could be improved. However, we made no effort to do that in order to keep the algorithm as simple as possible.

Tree decompositions are tightly related to families of minimal separators [36]. We start introducing the minimal separators used for building our quasi star-decomposition. Precisely, an *almost-clique* is a set with exactly two nonadjacent vertices. A set that is both an almost-clique and a minimal separator is termed an *almost-clique minimal separator*. All the almost-clique minimal separators of a graph can be computed in $\mathcal{O}(n^2m)$-time [11]. Furthermore, relationships between almost-clique minimal separators and the computation of treewidth have already been proved in [11]. We first prove basic properties of almost-clique minimal separators, for general graphs with treebreadth one and for $K_{3,3}$-minor-free graphs.

For this purpose, we will take use of an extra property of star-decompositions.

Definition 7. A star-decomposition $(T, \mathcal{X})$ of a graph G is said *saturated* if it is reduced and, for every $\{t, t'\} \in E(T)$, if $X_t \subseteq N_G[v_t]$ for some $v_t \in X_t$ then $X_{t'} \cap N_G[v_t] \subseteq X_t$.

Intuitively, in a saturated star-decomposition we can no longer add a vertex into some bag while keeping the property to be a tree decomposition of breadth one.

Proposition 8. *Every graph with treebreadth one admits a saturated star-decomposition.*

Proof. Note that given any star-decomposition $(T, \mathcal{X})$, the property can always be enforced by repeatedly taking a bag X_t, $t \in V(T)$, and a vertex $v_t \in X_t$ dominating this bag, then adding all of $\bigcup_{t' \in N_T(t)} X_{t'} \cap N_G[v_t]$ in X_t. □

Recall that, given a tree decomposition $(T, \mathcal{X})$ of a graph $G = (V, E)$, for any $v \in V$, T_v denotes the subtree of T induced by the bags containing v.

Lemma 10. *Let $G = (V, E)$ be a graph with $tb(G) = 1$. Every saturated star-decomposition $(T, \mathcal{X})$ of G has the following property. For every almost-clique minimal separator S of G, exactly one of the following holds for the unique pair of nonadjacent vertices $u, v \in S$:*

- $T_u \cap T_v \neq \emptyset$;
- *or there is* $\{t_u, t_v\} \in E(T)$ *such that:* $X_{t_u} \in T_u$, $X_{t_v} \in T_v$ *and* $X_{t_u} \subseteq N_G[u]$, $X_{t_v} \subseteq N_G[v]$.

Proof. Let $(T, \mathcal{X})$ be any saturated star-decomposition of G, that exists by Proposition 8. Let S be an almost-clique minimal separator and let $u, v \in S$ be nonadjacent. Suppose $T_u \cap T_v = \emptyset$ (otherwise we are already done). Let $(B_1, \cdots, B_r)$ be the shortest T_uT_v-path in T with $u \in B_1$, $v \in B_r$ and $\{u, v\} \cap B_i = \emptyset$ for every $1 < i < r$. Since $S \setminus \{u, v\} \subseteq N_G(u) \cap N_G(v)$, $S \setminus \{u, v\} \subseteq B_i$ for every $1 \leq i \leq r$. Moreover, since S is a minimal separator, every B_i, $1 \leq i \leq r$, must intersect at least two connected components of $G \setminus S$. Therefore, if $r > 2$, B_2 can only be dominated by a vertex in $S \setminus \{u, v\}$, that contradicts the fact that $(T, \mathcal{X})$ is saturated (since we could add u in B_2). As a result, $r = 2$. Similarly, if B_1 (resp., B_r) is not dominated by u (resp., by v), it must be dominated by a vertex in $S \setminus \{u, v\}$, that contradicts the fact that $(T, \mathcal{X})$ is saturated (we could add v in B_1, resp., u in B_r). Therefore, $B_1 \subseteq N_G[u]$ and $B_r \subseteq N_G[v]$ are two adjacent bags as stated in the lemma. □

In the case of $K_{3,3}$-minor-free graphs, almost-clique minimal separators have a very nice structure. Namely, they are either 2-separators (i.e., a separator of size 2) or they induce a path of length three. We prove it next.

Lemma 11. *Let $G = (V, E)$ be $K_{3,p}$-minor-free, let $v \in V$ and $S \subseteq N_G(v)$ be a minimal separator of $G \setminus v$. Then, $|S| \leq p - 1$.*

Proof. Let A, B be full components for S in $G \setminus v$. There is a $K_{3,|S|}$-minor for G with respective sides $\{A, B, v\}$ and S. Since G is $K_{3,p}$-minor-free by the hypothesis, we deduce that $|S| \leq p-1$. □

Corollary 9. *Let $G = (V, E)$ be a $K_{3,3}$-minor-free graph with $tb(G) = 1$. An almost-clique minimal separator of G is either a 2-separator or it induces a P_3.*

Proof. Let S be an almost-clique minimal separator of G. We have $|S| \geq 2$ since there are two nonadjacent vertices contained in S. In particular, $|S| \leq 2$ implies that S is a minimal 2-separator. Otherwise, $|S| \geq 3$, hence there exists $v \in S$ such that $S \subseteq N_G[v]$. Applying Lemma 11 to v and $S \setminus v$, one obtains $|S \setminus v| \leq 2$. Therefore, $|S| = 3$, *i.e.*, S induces a P_3 in this case. □

We are now ready to define the auxiliary tree decomposition that we use for the algorithm. Given two minimal separators S_1, S_2 of G, we say that S_1 *crosses* S_2 if S_1 intersects at least two components of $G \setminus S_2$. Two minimal separators that do not cross each other are called *parallel*. It is well known (*e.g.*, see [36]) that any family $\mathcal{S}$ of pairwise parallel minimal separators defines a tree decomposition. Roughly, such a tree decomposition can be computed recursively. If $\mathcal{S} = \emptyset$ then it suffices to compute a trivial tree decomposition with one node. Otherwise, let $S \in \mathcal{S}$ be arbitrary. We partition the connected components of $G \setminus S$ into sets $V_1, V_2, \ldots, V_k$ such that every set contains exactly one full component (connected components that are not full are all put together with an arbitrary full component). It takes $\mathcal{O}(m)$-time. Then, since S is parallel to every separator in $\mathcal{S} \setminus S$, for every $S' \in \mathcal{S} \setminus S$ and for every $1 \leq i \leq k$ either $S' \subseteq S \cup V_i$ or $S' \cap V_i = \emptyset$. Let $\mathcal{S}_i \subseteq \mathcal{S} \setminus S$ contain the elements of $\mathcal{S}$ that are minimal separators of $G_i = G[V_i \cup S]$. By induction on $|\mathcal{S}_i|$, for every $1 \leq i \leq k$, $\mathcal{S}_i$ defines a tree decomposition $(T^i, \mathcal{X}^i)$ of G_i, with one bag containing S (since S is parallel to every minimal separator in $\mathcal{S}_i$). Finally, we connect the tree decompositions $(T^1, \mathcal{X}^1), (T^2, \mathcal{X}^2), \ldots, (T^k, \mathcal{X}^k)$ into one tree decomposition $(T, \mathcal{X})$ by adding $k-1$ edges between the bags that contain S. These $k-1$ new edges $\{t, t'\}$ correspond to S in the sense that $X_t \cap X_{t'} = S$. Therefore the edges of T are in many-to-one correspondence with the separators in the family [36].

Definition 10. A *quasi star-decomposition* of G is a tree decomposition obtained from an inclusion wise maximal family of pairwise parallel almost-clique minimal separators. A *quasi bag* is a bag in a quasi star-decomposition.

The above procedure we sketched takes $\mathcal{O}(|\mathcal{S}|m)$-time in order to compute a tree decomposition from a family $\mathcal{S}$, that is in $\mathcal{O}(nm)$ since $\mathcal{S}$ only contains pairwise parallel minimal separators [6, 12, 36]. In particular, since all almost-clique minimal separators can be computed in $\mathcal{O}(n^2m)$-time for any graph, a quasi star-decomposition can also be computed in $\mathcal{O}(n^2m)$-time.

In what follows, we present a key property of quasi bags, that is the cornerstone of our algorithm.

Lemma 12. *Let $G = (V, E)$ be a prime $K_{3,3}$-minor-free graph with $tb(G) = 1$. Every saturated star-decomposition $(T, \mathcal{X})$ of G is such that, for every quasi-bag B, the family $(X_t \cap B)_{t \in V(T)}$ has at most two inclusionwise maximal elements.*

Proof. Let $(T, \mathcal{X})$ be any saturated star-decomposition of G, that exists by Proposition 8. Let $\mathcal{S}$ be an inclusion wise maximal family of pairwise parallel almost-clique minimal separators. For every $S \in \mathcal{S}$ with $u, v \in S$ nonadjacent and $T_u \cap T_v \neq \emptyset$ we add an edge $\{u, v\}$. By construction, $(T, \mathcal{X})$

is a star-decomposition of the resulting graph G'. Then, let B be a quasi-bag in the quasi star-decomposition obtained from $\mathcal{S}$. We keep the atom A of G' that contains B. Applying recursively Lemma 7 to the clique-separators that border A, $(T,(X_t \cap A)_{t\in V(T)})$ is a tree decomposition of both $G[A]$ and $G'[A]$ of breadth one. Let $(T',\mathcal{X}')$ be a reduced star-decomposition obtained from $(T,(X_t \cap A)_{t\in V(T)})$ (Lemma 1). We claim that T' has at most two nodes, that will prove the lemma.

Suppose by contradiction T' has at least three nodes. Let $t \in V(T')$ be an internal node. By the properties of a tree decomposition, X'_t is a separator of $G'[A]$, that disconnects all the vertices of $\left(\bigcup_{t_1\in V(T'_1)} X'_{t_1}\right) \setminus X'_t$ from all the vertices of $\left(\bigcup_{t_2\in V(T'_2)} X'_{t_2}\right) \setminus X'_t$ for every distinct subtrees T'_1, T'_2 of $T' \setminus t$. In what follows, we prove that X'_t contains an almost-clique minimal separator of G, that is not in $\mathcal{S}$ but is parallel to every separator in $\mathcal{S}$. The latter will arise a contradiction by maximality of $\mathcal{S}$. First let us prove the following subclaim:

Claim 1. *For every $S \in \mathcal{S}$, $S \subseteq A$, there exists a subtree T'_S of $T' \setminus t$ such that $S \subseteq X'_t \cup \left(\bigcup_{t_S\in V(T'_S)} X'_{t_S}\right)$.*

Proof. Let us show that there is one bag B or two adjacent bags B_1, B_2 such that $S \subseteq B$, resp. $S \subseteq B_1 \cup B_2$. The latter will prove the claim directly. Let $u, v \in S$ be nonadjacent. If $T'_u \cap T'_v \neq \emptyset$ then, since S is an almost-clique, by the Helly property $\bigcap_{s\in S} T'_s \neq \emptyset$, so, the subclaim is proved in this case. Otherwise, by Lemma 10 there are two adjacent bags respectively dominated by u and v. Since $S \setminus (u, v) \subseteq N_G(u) \cap N_G(v)$ we have that all of S is contained in the two adjacent bags, so, the subclaim is also proved in this case. $\diamond$

Then, let $H_{\mathcal{S}}$ be the supergraph of G' (and so, of G) obtained by completing every separator in $\mathcal{S}$ into a clique. Observe that the atoms of $H_{\mathcal{S}}$ are exactly the quasi bags of the quasi star-decomposition obtained from $\mathcal{S}$ (since G is prime). In particular, B is an atom of $H_{\mathcal{S}}$. According to the previous claim, for every distinct subtrees T'_1, T'_2 of $T' \setminus t$ there is no edge added in $H_{\mathcal{S}}$ between $\left(\bigcup_{t_1\in V(T'_1)} X'_{t_1}\right) \setminus X'_t$ and $\left(\bigcup_{t_2\in V(T'_2)} X'_{t_2}\right) \setminus X'_t$. Hence, X'_t keeps the property to be a separator in $H_{\mathcal{S}}$. In particular, it is a separator of $H_{\mathcal{S}}[A']$ for some atom $A' \subseteq A$ (possibly, $B = A'$). In this situation, let $v_t \in X'_t$ such that $X'_t \subseteq N_G[v_t]$, that exists since X'_t is a bag in a star-decomposition of $G[A]$. The subgraph $H_{\mathcal{S}}[A']$ is a contraction-minor of G. Indeed, it can be obtained from G by contracting, for every $S \in \mathcal{S}$ that borders A', with $u, v \in S$ nonadjacent, the set W of all connected components C of $G \setminus S$, $C \subseteq V \setminus A'$, into a single node representing $W \cup \{u\}$. Since, by minimality of the separators in $\mathcal{S}$, at least one such W is a full component, it "creates" the missing edge between u and v in $H_{\mathcal{S}}$. In particular, $H_{\mathcal{S}}[A']$ is $K_{3,3}$-minor-free. Thus, we can apply Lemma 11 to the subgraph $H_{\mathcal{S}}[A'] \setminus v_t$ (that is connected, since $H_{\mathcal{S}}[A']$ is prime) in order to extract from $X'_t \setminus v_t$ a minimal separator $S^c \subset N[v_t]$ of the subgraph $H_{\mathcal{S}}[A']$, $|S^c| \leq 2$. In particular, S^c or $S^c \cup \{v_t\}$ is a minimal separator of $H_{\mathcal{S}}[A']$, and in both cases it is an almost-clique minimal separator since $H_{\mathcal{S}}[A']$ is prime, $|S^c| \leq 2$ and $S^c \subset N_G[v_t]$. By construction, this above minimal separator is an almost-clique of G. Altogether combined, there exists an almost-clique minimal separator of $H_{\mathcal{S}}[A']$, and so, an almost-clique minimal separator of G that is not in $\mathcal{S}$ but is parallel to every separator in $\mathcal{S}$ (since all separators in $\mathcal{S}$ have been completed into cliques and so, cannot disconnect an atom of $H_{\mathcal{S}}$ nor cross a separator of such an atom). The latter arises a contradiction by maximality of $\mathcal{S}$, hence it proves the claim. As a result, the reduced star-decomposition $(T', \mathcal{X}')$ is such T' has at most two nodes. $\square$

We finally prove Theorem 6. For this purpose, we use the following notations. Let $(\langle T, r\rangle, \mathcal{X})$

be a tree decomposition of G where T is rooted in $r \in V(T)$. For every $t \in V(T)$, let T_t be the subtree of $\langle T, r \rangle$ rooted at t. Let $V_t = \bigcup_{t' \in V(T_t)} X_{t'}$, and let $G_t = G[V_t]$.

Proof of Theorem 6. In what follows, we describe our algorithm and prove its correctness.

First, a quasi star-decomposition $(T, \mathcal{X})$ of G is computed in time $\mathcal{O}(n^2m)$ from any inclusionwise maximal family $\mathcal{S}$ of pairwise parallel almost-clique minimal separators. According to Definition 10, every edge $e = \{t, t'\} \in E(T)$ is associated to an almost-clique minimal separator $S_e = X_t \cap X_{t'} \in \mathcal{S}$ of G. For every edge $e = \{t, t'\} \in E(T)$, let $u_e, v_e \in S_e$ be the unique pair of nonadjacent vertices in S_e.

Let T be rooted in an arbitrary vertex r.

First we consider the particular case when there is only one quasi-bag, *i.e.*, $V(T) = \{r\}$. According to Proposition 8, there exists a saturated star-decomposition of G (otherwise, $tb(G) > 1$). Furthermore, since $V = X_r$ is a quasi-bag, by Lemma 12 there can be no more than two bags in a saturated star-decomposition of G. Hence, the algorithm checks whether G contains a universal vertex (star-decomposition with one bag) or whether there exist $u, v \in V$ such that $N_G[u], N_G[v]$ are the two bags of a tree decomposition of G (saturated star-decomposition with two bags). If not, then we conclude $tb(G) > 1$. It takes $\mathcal{O}(n^2m)$-time.

From now on, we assume there are at least two quasi-bags. For every $t \in V(T) \setminus \{r\}$, let e_t be the parent-edge incident to t (i.e., the edge incident to t on the shortest path from t to r), and let e_r be any edge incident to r. The algorithm proceeds by dynamic programming, bottom-up from the leaves of T to the root r and recursively computes, for every $t \in V(T)$:

Type A_t. a star-decomposition of G_t with some bag containing S_{e_t} (if such a decomposition exists), and

Type B_t. a star-decomposition of G_t with two adjacent bags B_1 and B_2 such that $B_1 \subseteq N_G[u_{e_t}]$ and $B_2 \subseteq N_G[v_{e_t}]$ (if such a decomposition exists).

Clearly, if such a decomposition (Type A_r or B_r) is computed for $G_r = G$, then $tb(G) = 1$. In what follows, we describe how the algorithm proceeds and prove that, if for some vertex $t \in V(T)$ no such decompositions are found (neither of Type A_t, nor of Type B_t), then we can conclude that $tb(G) > 1$.

Let $t \in V(T)$ and let $\mathcal{S}$ be the set of all almost-clique minimal separators corresponding to some edge incident to t. The algorithm proceeds in two phases.

- During the first phase, the algorithm checks whether the graph $G[X_t]$ induced by the quasi-bag X_t admits some star-decompositions with specific properties defined below.

  Let $S \in \mathcal{S}$ be an almost-clique minimal separator corresponding to an edge incident to t, and let u_S and v_S be the two nonadjacent vertices of S. A star-decomposition of $G[X_t]$ is of Type S if it satisfies the following two conditions:

  - it consists of two bags $N[u_S] \cap X_t$ and $N[v_S] \cap X_t$, and
  - for every almost-clique minimal separator $S' \in \mathcal{S}$, with u'_S and v'_S being the two nonadjacent vertices of S', if $\{u'_S, v'_S\} \neq \{u_S, v_S\}$, then u'_S and v'_S are contained in a same bag.

  A star-decomposition of $G[X_t]$ is said to be of Type E if:

– it consists of at most two bags, and
– for every almost-clique minimal separator $S \in \mathcal{S}$, with u_S and v_S being the two nonadjacent vertices of S, then u_S and v_S are contained in a same bag.

The goal of this phase is, for every almost-clique minimal separator $S \in \mathcal{S}$, to compute a star-decomposition of $G[X_t]$ of Type S (if any), and to compute a star-decomposition of $G[X_t]$ of Type E (if any).

We first show that, if $tb(G) = 1$, then $G[X_t]$ must admit at least one such a decomposition (of type E or of Type S for some almost-clique minimal separator $S \in \mathcal{S}$), then we show how to compute them in polynomial-time.

Claim 2. *If $tb(G) = 1$, then $G[X_t]$ admits a star-decomposition of Type E or of Type S for some $S \in \mathcal{S}$.*

Proof. If $tb(G) = 1$ then let $(T^*, \mathcal{Y}^*)$ be any saturated star-decomposition of G (that exists by Proposition 8). Let $(T_t^*, \mathcal{Y}_t^*)$ be a reduced tree decomposition of $G[X_t]$ of breadth one, obtained from $(T^*, \{Y \cap X_t \mid Y \in \mathcal{Y}^*\})$ (Lemma 1). By Lemma 12, $(T_t^*, \mathcal{Y}_t^*)$ has at most two bags.

If $(T_t^*, \mathcal{Y}_t^*)$ is of Type E then we are done. Otherwise, it means that there is an almost-clique minimal separator $S \in \mathcal{S}$, with its nonadjacent vertices u_S and v_S that are not in a same bag. We prove in what follows that $(T_t^*, \mathcal{Y}_t^*)$ is of Type S. By construction of $(T_t^*, \mathcal{Y}_t^*)$, u_S and v_S are not in a same bag of $(T^*, \mathcal{Y}^*)$. Therefore, by Lemma 10, $(T^*, \mathcal{Y}^*)$ has two adjacent bags $X \subseteq N_G[u_S]$ and $Y \subseteq N_G[v_S]$. Since $(T_t^*, \mathcal{Y}_t^*)$ is obtained by reducing $(T^*, \mathcal{Y}^* \cap X_t)$, it can only be that $(T_t^*, \mathcal{Y}_t^*)$ consists exactly of two bags B'_u, B'_v, that contain $N_G[u_S] \cap X_t$ and $N_G[v_S] \cap X_t$, respectively. Furthermore, $Z = N_G(u_S) \cap N_G(v_S) \cap X_t = B'_u \cap B'_v$ is a separator of $G[X_t]$. Suppose by contradiction that $B'_u \not\subseteq N[u]$. Let $x, x' \in B'_u \setminus u$ be such that $B'_u \subseteq N[x]$ and $x' \notin N[u]$. We first observe that $Z \cup \{u\}$ is an $x'v$-separator of $G[X_t]$, that is parallel with every minimal separator of $\mathcal{S}$ by construction. Therefore, $|Z| \geq 3$ (otherwise, $Z \cup \{u\}$ would be either a clique or an almost-clique). Since we also have a $K_{3,|Z \setminus \{x\}|}$-minor with respective sides $\{u, v, x\}$ and $Z \setminus \{x\}$, we get $x \in Z$ and $|Z| = 3$. But then again, this implies that $Z \cup \{u\}$ is an almost-clique, a contradiction. As a result, the two bags B'_u, B'_v, are exactly $N_G[u_S] \cap X_t$ and $N_G[v_S] \cap X_t$.

Finally, let us assume by contradiction that there is an almost-clique minimal separator $S' \in \mathcal{S}$, $S' \neq S$, with u'_S and v'_S being the two nonadjacent vertices of S', $\{u'_S, v'_S\} \neq \{u_S, v_S\}$, and u'_S and v'_S are not contained in a same bag of $(T_t^*, \mathcal{Y}_t^*)$. W.l.o.g., u_S, u'_S are contained in a common bag, v_S, v'_S are contained in a common bag, and $u_S \neq u'_S$. In this situation, since S and S' play symmetric roles, the only two bags of $(T_t^*, \mathcal{Y}_t^*)$ are dominated by both u_S, u'_S and by v_S, v'_S, respectively. By construction, the same holds for two adjacent bags in $(T^*, \mathcal{Y}^*)$. Then, $u_S, v_S \in S'$ since they have neighbours in two full components for $G \setminus S'$. However, in this situation there are at least two non edges in $G[S']$, thereby contradicting that S' is a quasi-clique.

Therefore, $(T_t^*, \mathcal{Y}_t^*)$ is of Type S. $\diamond$

In order to check whether the desired decompositions exists, since they have at most two bags, it is sufficient to check which of the following conditions hold:

– there exists a star-decomposition of G_t with a unique bag, equivalently G_t admits a dominating vertex.

– for every $u, v \in X_t$, there is a star-decomposition of G_t with its two bags being respectively dominated by u and v; equivalently, we check whether $(N_G[u] \cap X_t, N_G[v] \cap X_t)$ is a star-decomposition of $G[X_t]$.

Clearly, if a star-decomposition of Type E (resp., of Type S) exists for $G[X_t]$, the above procedure computes it. Therefore, if no such decomposition is computed for $G[X_t]$, then $tb(G) > 1$.

Overall, it takes $\mathcal{O}(|X_t|^2|E(G[X_t])|) = \mathcal{O}(n^2m)$-time.

Note that, it t is a leaf, a star-decomposition of Type E is of Type A_t. Indeed, $S_{e_t} \setminus \{u_{e_t}, v_{e_t}\}$ is a clique and a common neighborhood of u_{e_t} and v_{e_t}, so, by the Helly property it must be contained in a common bag with u_{e_t} and v_{e_t}. Similarly, if $S \in \mathcal{S} \setminus \{S_{e_t}\}$ satisfies $\{u_S, v_S\} \neq \{u_{S_{e_t}}, v_{S_{e_t}}\}$, then a star-decomposition of Type S is of Type A_t. Moreover, a star-decomposition of Type S_{e_t} is of Type B_t. In the same way, if $S \in \mathcal{S} \setminus \{S_{e_t}\}$ satisfies $\{u_S, v_S\} = \{u_{S_{e_t}}, v_{S_{e_t}}\}$, then a star-decomposition of Type S is of Type B_t. Therefore, if t is a leaf, then after the first phase of the algorithm, we are done.

Otherwise, the second phase of the algorithm aims at combining the decomposition(s) computed for $G[X_t]$ with the one(s) computed recursively for the children of t.

- Let us assume that t is not a leaf and let $t_1, \cdots, t_d$ be the children of t. By dynamic programming, we suppose for every $1 \leq i \leq d$, the algorithm has already computed a star-decomposition of Type A_{t_i} and/or a star-decomposition of Type B_{t_i} of the graph G_{t_i} (if no such decomposition exists, then by the induction hypothesis, we have already concluded that $tb(G) > 1$). In what follows, we explain how the algorithm will combine these decompositions with the ones of $G[X_t]$ computed during the first phase.

  1. Let us assume first that there exists an almost-clique minimal separator $S \in \mathcal{S}$ such that:

     – the algorithm in the first phase has computed a star-decomposition of Type S for $G[X_t]$, and

     – for every edge $e_i = \{t, t_i\}$ incident to t and such that $\{u_S, v_S\} = \{u_{S_{e_i}}, v_{S_{e_i}}\}$, then the dynamic programming has computed a star-decomposition of Type B_{t_i} for G_{t_i}, and

     – for every other edge $e_i = \{t, t_i\}$ incident to t (i.e., $\{u_S, v_S\} \neq \{u_{S_{e_i}}, v_{S_{e_i}}\}$), then the dynamic programming has computed a star-decomposition of Type A_{t_i} for G_{t_i}.

     Recall that the decomposition of $G[X_t]$ consists of the two bags $N[u_S] \cap X_t$ and $N[v_S] \cap X_t$, and that, for every almost-clique minimal separator $S' \in \mathcal{S}$, $\{u_S, v_S\} \neq \{u_{S'}, v_{S'}\}$, then S' is fully included in some bag (since $u_{S'}$ and $v_{S'}$ are in a same bag and because $S' \setminus \{u_{S'}, v_{S'}\}$ is a clique in the common neighborhood of $u_{S'}$ and $v_{S'}$).

     W.l.o.g., let us assume that $\{u_S, v_S\} = \{u_{S_{e_i}}, v_{S_{e_i}}\}$ for every $1 \leq i \leq j$, for some $j \leq d$ (in the case when we only have $\{u_S, v_S\} = \{u_{S_{e_t}}, v_{S_{e_t}}\}$, with e_t being the edge between t and its parent node, there is nothing to do). For every $1 \leq i \leq j$, the star-decomposition (of Type B_{t_i}) of G_{t_i} has two adjacent bags $B_u^i \subseteq N_G[u_S]$ and $B_v^i \subseteq N_G[v_S]$. We combine all these decompositions and the one of $G[X_t]$ by merging the bags $N[u_S] \cap X_t, B_u^1, \cdots, B_u^j$ and merging the bags $N[v_S] \cap X_t, B_v^1, \cdots, B_v^j$. Let $(Y, \mathcal{Z})$ be the resulting decomposition. Finally, for every $j < i \leq d$, we add an edge between a bag of the decomposition of G_{t_i} that contains S_{e_i} (such a bag exists since this decomposition is of Type A_{t_i}), and a bag of the decomposition $(Y, \mathcal{Z})$ that contains S_{e_i} (that exists by the remark above).

The obtained decomposition is clearly a star-decomposition of G_t. Moreover, it is of Type A_t if $\{u_S, v_S\} \neq \{u_{S_{e_t}}, v_{S_{e_t}}\}$ and of Type B_t otherwise.

2. The second case to be considered is when:
   - the algorithm in the first phase has computed a star-decomposition of Type E for $G[X_t]$, and
   - for every edge $1 \leq i \leq d$, then the dynamic programming has computed a star-decomposition of Type A_{t_i} for G_{t_i}.

   For every $1 \leq i \leq d$, we add an edge between a bag of the decomposition of G_{t_i} that contains S_{e_i} (such a bag exists since this decomposition is of Type A_{t_i}), and a bag of the decomposition of $G[X_t]$ that contains S_{e_i} (that exists by the remark above).

   The obtained decomposition is clearly a star-decomposition of Type A_t of G_t.

It only remains to prove that, if $tb(G) = 1$, we must be in a case as above (and therefore, if no such decomposition can be created, we can conclude that $tb(G) > 1$). Indeed, assume $tb(G) = 1$ and let $(T^*, \mathcal{Y}^*)$ be any saturated star-decomposition of G (that exists by Proposition 8). Let $S \in \mathcal{S}$. By Lemma 10, $(T^*, \mathcal{Y}^*)$ either has a bag containing S, or it has two adjacent bags $X \subseteq N_G[u_S]$ and $Y \subseteq N_G[v_S]$.

Let us first consider the latter case. For every $e_i = \{t, t_i\}$ such that $\{u_S, v_S\} = \{u_{S_{e_i}}, v_{S_{e_i}}\}$, restricting $(T^*, \mathcal{Y}^*)$ to the vertices of G_{t_i} leads to a star-decomposition of Type B_{t_i}. Moreover, by the proof of Claim 2, the decomposition $(T_t^*, \mathcal{Y}_t^*)$ of $G[X_t]$ obtained from $(T^*, \mathcal{Y}^*)$ is of Type S. Therefore, for every $e_i = \{t, t_i\}$ such that $\{u_S, v_S\} \neq \{u_{S_{e_i}}, v_{S_{e_i}}\}$, some bag of $(T_t^*, \mathcal{Y}_t^*)$ contains S_{e_i}. Hence, it is also the case for the decomposition obtained by restricting $(T^*, \mathcal{Y}^*)$ to the vertices of G_{t_i}. In other words, such a decomposition is of Type A_{t_i}. Overall, we are in the first case.

Second, assume that for every $S \in \mathcal{S}$, $(T^*, \mathcal{Y}^*)$ has a bag containing S. Then, restricting $(T^*, \mathcal{Y}^*)$ to the vertices of X_t leads to a decomposition of $G[X_t]$ of Type E, and, for every $1 \leq i \leq d$, restricting $(T^*, \mathcal{Y}^*)$ to the vertices of G_{t_i} leads to a decomposition of G_{t_i} of Type A_{t_i}, i.e., we are in the second case.

□

If follows from [16] that $K_{3,3}$-minor-free graphs with bounded treebreadth also have bounded treewidth. However, there is no explicit upper-bound given. Before concluding this section, we improve the upper-bound on treewidth for $K_{3,3}$-minor-free graphs with unit treebreadth.

Corollary 11. *For every $K_{3,3}$-minor-free graph G with $tb(G) = 1$ we have $tw(G) \leq 4$, and the upper-bound is sharp.*

Proof. The treewidth of a graph is the maximum treewidth over the subgraphs induced by its atoms [11]. By Lemma 7, $tb(G) = 1$ implies that every subgraph induced by an atom of G also has treebreadth one. Thus, we can assume G to be prime.

Furthermore, if S is an almost-clique minimal separator of G then we can bipartition $V(G)$ into two disjoint sets A, B such that $A \cap B = S$ and both A and B contain a full component for S. Let G_A, G_B be, respectively, obtained from $G[A]$ and $G[B]$ by adding an edge between the unique pair of nonadjacent $u, v \in S$. Since G_A and G_B are contraction-minors of G we have that G_A, G_B are $K_{3,3}$-minor-free, $tw(G) = \max\{tw(G_A), tw(G_B)\}$ [11] and $tb(G_A) = tb(G_B) = 1$. Let us repeat these two above operations until all the subgraphs obtained are prime and with no almost-clique minimal separator. Doing so, we can assume from now on G to be prime and with no almost-clique minimal separator.

Let $(T, \mathcal{X})$ be a star-decomposition of G. Observe that any tree decomposition $(T', \mathcal{Y}')$ such that, for every $t' \in V(T')$, $Y'_{t'} \subseteq X_t$ for some $t \in V(T)$, also has treebreadth equal to one. So, let us assume $(T, \mathcal{X})$ to be *minimal*, *i.e.*, there is no such (reduced) tree decomposition as above. Minimal tree decompositions have been characterized in [12]. We can prove, the same way as for Lemma 12, there are at most two bags in the decomposition (otherwise, we could extract an almost-clique minimal separator from an internal bag of the star-decomposition). According to [12], there are two cases.

- Suppose there is only one bag. Then, G is a complete graph [12]. Since G is $K_{3,3}$-minor-free, it has order at most 5. Hence, $tw(G) \leq 4$.
- Otherwise, let $V(T) = \{t, t'\}$ and let $u, v \in V(G)$. The subsets $K_t = X_t \setminus X_{t'}$, $K_{t'} = X_{t'} \setminus X_t$ are dominating cliques of $G[X_t]$ and $G[X_{t'}]$, respectively, while $S = X_t \cap X_{t'}$ is a minimal separator [12]. In particular, $|S| \geq 3$ since G is prime and it has no almost-clique minimal separators. Since G is $K_{3,3}$-minor-free it implies $|K_t| + |K_{t'}| \leq 2$, and so, $K_t, K_{t'}$ are both reduced to a singleton. Furthermore we claim that $G[S]$ has maximum degree two. Indeed, for every $x \in S$, there is a $K_{3,deg_{G[S]}(x)}$-minor with respective sides $\{x, K_t, K_{t'}\}$ and $N_{G[S]}(x)$. The latter proves the claim since G is $K_{3,3}$-minor-free. As a result, $G[S]$ is a disjoint union of cycles and paths. It implies $tw(G[S]) \leq 2$, thus $tw(G) \leq tw(G[S]) + |V(G) \setminus S| \leq 4$.

The upper-bound is reached by K_5 and by the planar graph H_4, obtained from a C_4 by adding two nonadjacent vertices $u, v \notin V(C_4)$ of degree four. □

4 Conclusion.

We conclude this paper by some questions that remain open. First, it would be interesting to know the complexity of computing the treebreadth of planar graphs. Second, all the reductions presented in this paper rely on constructions containing large clique or clique-minor. We left open the problem of recognizing graphs with treebreadth one in the class of graphs with bounded treewidth or bounded clique-number. More generally, is the problem of computing the treebreadth Fixed-Parameter Tractable when it is parameterized by the treewidth or by the size of a largest clique-minor?

References

[1] M. Abu-Ata and F. Dragan. Metric tree-like structures in real-world networks: an empirical study. *Networks*, 67:49–68, 2016.

[2] J. Andrade, H. Herrmann, R. Andrade, and L. da Silva. Apollonian networks: Simultaneously scale-free, small world, euclidean, space filling, and with matching graphs. *Phys. Rev. Lett.*, 94, 2005.

[3] S. Arnborg, D. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

[4] R. Belmonte, F. V. Fomin, P. A. Golovach, and M. S. Ramanujan. Metric dimension of bounded tree-length graphs. *SIAM J. Discrete Math.*, 31(2):1217–1243, 2017.

[5] A. Berry, R. Pogorelcnik, and A. Sigayret. Vertical decomposition of a lattice using clique separators. In *CLA'11, Nancy, France*, pages 15–29, 2011.

[6] A. Berry, R. Pogorelcnik, and G. Simonet. An introduction to clique minimal separator decomposition. *Algorithms*, 3(2):197–215, 2010.

[7] H. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[8] H. Bodlaender. Treewidth: Characterizations, applications, and computations. In *WG 2006, Bergen, Norway*, pages 1–14, 2006.

[9] H. Bodlaender, M. Fellows, and T. Warnow. Two strikes against perfect phylogeny. In *ICALP'92, Vienna, Austria*, pages 273–283, 1992.

[10] H. Bodlaender and T. Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21(2):358–402, 1996.

[11] H. L. Bodlaender and A. Koster. Safe separators for treewidth. *Discrete Mathematics*, 306(3):337–350, 2006.

[12] V. Bouchitté and I. Todinca. Listing all potential maximal cliques of a graph. *Theoretical Computer Science*, 276(1):17–32, 2002.

[13] A. Brandstädt, F. Dragan, V. Chepoi, and V. Voloshin. Dually chordal graphs. *SIAM Journal on Discrete Mathematics*, 11(3):437–455, 1998.

[14] S. Chechik, D. Larkin, L. Roditty, G. Schoenebeck, R. Tarjan, and V. Vassilevska Williams. Better approximation algorithms for the graph diameter. In *ACM SODA'14*, pages 1041–1052. SIAM, 2014.

[15] V. Chepoi, F. Dragan, B. Estellon, M. Habib, and Y. Vaxès. Diameters, centers, and approximating trees of δ-hyperbolic geodesic spaces and graphs. In *SCG '08*, pages 59–68, New York, NY, USA, 2008. ACM.

[16] D. Coudert, G. Ducoffe, and N. Nisse. To approximate treewidth, use treelength! *SIAM Journal of Discrete Mathematics*, 30(3):1424–1436, 2016.

[17] F. de Montgolfier, M. Soto, and L. Viennot. Treewidth and hyperbolicity of the internet. In *Network Computing and Applications (NCA), 2011 10th IEEE International Symposium on*, pages 25–32, Aug 2011.

[18] Y. Dourisboure, F. Dragan, C. Gavoille, and Y. Chenyu. Spanners for bounded tree-length graphs. *Theor. Comput. Sci.*, 383(1):34–44, 2007.

[19] Y. Dourisboure and C. Gavoille. Tree-decompositions with bags of small diameter. *Discrete Mathematics*, 307(16):2008–2029, 2007.

[20] F. Dragan and M. Abu-Ata. Collective additive tree spanners of bounded tree-breadth graphs with generalizations and consequences. *Theoretical Computer Science*, 547:1–17, 2014.

[21] F. Dragan and E. Köhler. An approximation algorithm for the tree t-spanner problem on unweighted graphs via generalized chordal graphs. *Algorithmica*, 69(4):884–905, 2014.

[22] F. Dragan, E. Köhler, and A. Leitert. Line-distortion, bandwidth and path-length of a graph. In *Algorithm Theory–SWAT 2014*, pages 158–169. Springer, 2014.

[23] F. Dragan and A. Leitert. On the minimum eccentricity shortest path problem. In *Algorithms and Data Structures – WADS*, pages 276–288. Springer, 2015.

[24] G. Ducoffe. A short note on the complexity of computing strong pathbreadth. Submitted to IPL.

[25] G. Ducoffe, N. Legay, and N. Nisse. On the complexity of computing treebreadth. In *IWOCA 2016 – 27th International Workshop on Combinatorial Algorithms*, pages 3–15, 2016.

[26] G. Ducoffe, S. Legay, and N. Nisse. On computing tree and path decompositions with metric constraints on the bags. Technical Report RR-8842, Inria, 2016.

[27] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, 16(1):47–56, 1974.

[28] M. Golumbic, H. Kaplan, and R. Shamir. On the complexity of dna physical mapping. *Advances in Applied Mathematics*, 15(3):251–261, 1994.

[29] M. C. Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57. Elsevier, 2004.

[30] R. Krauthgamer and J. Lee. Algorithms on negatively curved spaces. In *FOCS'06*, pages 119–132. IEEE, 2006.

[31] A. Leitert. 3-colouring for dually chordal graphs and generalisations. *Information Processing Letters*, 128:21–26, 2017.

[32] A. Leitert and F. Dragan. On strong tree-breadth. In *International Conference on Combinatorial Optimization and Applications*, pages 62–76. Springer, 2016.

[33] D. Lokshtanov. On the complexity of computing treelength. *Discrete Applied Mathematics*, 158(7):820–827, 2010.

[34] G. Mertzios and P. Spirakis. Algorithms and almost tight results for 3-colorability of small diameter graphs. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 332–343. Springer, 2013.

[35] J. Opatrny. Total ordering problem. *SIAM J. Comput.*, 8(1):111–114, 1979.

[36] A. Parra and P. Scheffler. Characterizations and algorithmic applications of chordal graph embeddings. *Discrete Applied Mathematics*, 79(1):171–188, 1997.

[37] N. Robertson and P. Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.

[38] C. Wang, T. Liu, W. Jiang, and K. Xu. Feedback vertex sets on tree convex bipartite graphs. In *COCOA 2012, Banff, AB, Canada*, pages 95–102, 2012.

[39] Y. Wu, P. Austrin, T. Pitassi, and D. Liu. Inapproximability of treewidth and related problems. *J. Artif. Intell. Res. (JAIR)*, 49:569–600, 2014.